Improving Security and Privacy in Current Mobile Systems

Submitted by

Mohammed Shafiul Alam Khan

for the degree of Doctor of Philosophy

at

Royal Holloway, University of London

2017

Declaration

I, Mohammed Shafiul Alam Khan, hereby declare that this thesis and the work presented in it is entirely my own. Where I have consulted the work of others, this is always clearly stated.

Signed ..... (Mohammed Shafiul Alam Khan)
Date:

To my parents — Shajahan Ahamed Khan and Jobeda Akhter Khanom

Abstract

As has been widely discussed, the GSM system only offers unilateral authentication of the mobile phone to the network; this limitation permits a range of attacks. While adding support for mutual authentication would be highly beneficial, changing the way GSM serving networks operate is not practical.

The 3G and 4G mobile systems rectify the GSM weakness by providing mutual authentication between phone and network, and significantly improve their security properties by comparison with 2G (GSM). However, significant shortcomings remain with respect to user privacy, most notably the decades-old privacy problem of disclosure of the permanent subscriber identity (IMSI), a problem arising in all generations of mobile networks and that makes IMSI catchers a real threat. Although a number of possible modifications to 2G, 3G and 4G protocols have been proposed designed to provide greater user privacy, they all require significant alterations to the existing deployed infrastructures, which are almost certainly impractical in practice.

In this thesis we investigate whether it is possible to improve the security and privacy properties of the current mobile systems without changing the deployed infrastructure, i.e. the serving networks and mobile phones.

We describe a novel modification to the relationship between a subscriber identity module (SIM) and its home network which allows mutual authentication without affecting the GSM infrastructure, including the phones; the only necessary changes are to the authentication centres and the SIMs.

We further propose novel authentication schemes for 3G and 4G systems to defeat IMSI catchers. Our first scheme makes use of multiple IMSIs for an individual USIM to offer a degree of pseudonymity for a user. The second scheme prevents disclosure of the subscriber's IMSI by using a dynamic pseudo-IMSI that is only identifiable by the subscriber's home network. A major challenge in using pseudonymous IMSIs is possible loss of identity synchronisation between a USIM and its home network, an issue that has not been adequately addressed in earlier work. We present an approach for identity recovery to be used in the event of pseudo-IMSI desynchronisation. Both schemes require changes to the home network and the USIM, both owned by a single entity in the mobile systems, but not to the serving network, mobile phone or other internal network protocols, enabling simple, transparent and evolutionary migration. We provide analyses of the schemes, and verify their correctness and security properties using ProVerif.

Acknowledgement

First and foremost, I would like to express my deepest gratitude to Professor Chris J Mitchell, my PhD supervisor and role model. I have greatly benefited from his guidance, kindness, patience, and cordial support. I wish to say a heartfelt thank you to him. Indeed, this thesis would never have become reality without his insightful ideas, invaluable comments and precious feedback.

I am extremely grateful to my mother, Mrs Jobeda Akhter Khanom, my brothers, Mr Nura Alam Khan and Mr Ashraful Alam Khan, my aunt, Mrs Shamsunnahar Khanom and all of my family members and friends for their endless support, continuous endorsement and enlightening advice; to them all I wish to say a sincere thank you.

I am profoundly appreciative to my colleagues, Mr Wanpeng Li, Mr Po-Wah Yau, Ms Mwawi Nyirenda Kayuni, Dr Zhang Xiao, Mrs Fatma Al Maqbali, Mr Nasser Al-Fannah, and Mr Zhaoyi Fan, for the friendly atmosphere they have created for our reading group and for the invaluable feedback and insightful ideas I got from them.

Thank you to all my friends within the Royal Holloway, University of London. Thanks to my friend, Mr Mohammed Marbin, for his generous support during my PhD study. I would also like to express my gratitude to my colleagues at Institute of Information Technology, University of Dhaka, Bangladesh for their supports and valuable advices.

I would like to thank my sponsor, Commonwealth Scholarship Commission in the United Kingdom, for financial support. I believe that it would not be possible to complete this study without their generous support.

I am particularly grateful to my wife and my lovely kids for their support during the stressful time of PhD research. Their presence makes me happy and helps me to concentrate in my work.

Finally, I own my sincere thanks to my mother for giving me the strength and motivation to do a PhD. I am thankful to almighty for being able to complete this thesis.

Contents

List of Figures

List of Tables

List of Algorithms

List of Listings

List of Notations

$\|$	Concatenation
$\oplus$	Boolean exclusive-Or
A3	GSM authentication function
A5/1	GSM encryption algorithm # 1
A5/2	GSM encryption algorithm # 2
A5/3	GSM encryption algorithm # 3
A8	GSM key generation function
AK	Anonymity key
AMF	Authentication management field
AUTM	Authentication token in pseudo-IMSI resynchronisation
AUTN	Authentication token
AUTS	Authentication token in SQN resynchronisation
c2	3GPP conversion function
c3	3GPP conversion function
CK	3GPP cipher key
EK	Encryption key in the modifiable multiple IMSIs scheme
f1	3GPP network authentication function
f1*	3GPP resynchronisation message authentication function
f2	3GPP user authentication function
f3	3GPP cipher key derivation function
f4	3GPP integrity key derivation function
f5	3GPP anonymity key derivation function for normal operation
f5*	3GPP anonymity key derivation function for resynchronisation
f5**	Mask key derivation function for the robust pseudo-IMSIs scheme
f5***	Mask key derivation function for the robust pseudo-IMSIs scheme
IK	3GPP integrity key
K	Subscriber authentication key
K_a	Shared/derived secret key in modified GSM

K_{ASME}	Local master key in EPS
K_c	Cipher key in GSM
K_e	Shared secret key in the van den Broek et al. scheme
MAC	Message authentication code
MAC-M	Message authentication code for pseudo-IMSI resynchronisation
MAC-S	Message authentication code for SQN resynchronisation
RAND	Random challenge
RES	Authentication response
SMAC	Sequence-MAC in the predefined/modifiable multiple IMSIs schemes
SQN	Sequence number
SQN_{MS}	Sequence number suggested by mobile station
SRES	Signed response in GSM
XMAC	Expected MAC
XOR	Boolean exclusive-Or
XRES	Expected RES

List of Abbreviations

1G	First Generation
2G	Second Generation
3G	Third Generation
3GPP	3rd Generation Partnership Project
4G	Fourth Generation
ADF	Application Dedicated File
AKA	Authentication and Key Agreement
APDU	Application Protocol Data Unit
AuC	Authentication Center
AV	Authentication Vector
BCCH	Broadcast Common Control Channel
BS	Base Station
BSC	Base Station Controller
BSS	Base Station Subsystem
BTS	Base Transceiver Station
CC	Country Code
CEPT	European Conference of Postal and Telecommunications Administrations
CRL	Certificate Revocation List
CS	Circuit Switched
DF	Dedicated File
DMSI	Dynamic Mobile Subscriber Identity
EF	Elementary File
EIR	Equipment Identity Register
eNB	Evolved Node B
EPS	Evolved Packet System
ETSI	European Telecommunications Standard Institute
E-UTRAN	Evolved UTRAN
GERAN	GSM/EDGE Radio Access Network

GGSN	Gateway GPRS Support Node
GMSC	Gateway-MSC
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
GSMA	GSM Association
GUMMEI	Globally Unique MME Identifier
GUTI	Globally Unique Temporary UE Identity
HLR	Home Location Register
HN	Home Network
HSS	Home Subscriber Server
IMAN	International Mobile Anonymous Number
IMEI	International Mobile Equipment Identity
IMEISV	International Mobile Equipment Identity Software Version
IMS	IP Multimedia Subsystem
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
ISIM	IMS SIM
KDF	Key Derivation Function
LA	Location Area
LAC	Location Area Code
LAI	Location Area Identity
LTE	Long-Term Evolution
MAC	Message Authentication Code
MCC	Mobile Country Code
ME	Mobile Equipment
MF	Master File
MME	Mobility Management Entity
MNC	Mobile Network Code
MNO	Mobile Network Operator
MS	Mobile Station
MSC	Mobile Switching Center
MSIN	Mobile Subscriber Identification Number
MSISDN	Mobile Station International ISDN Number
MT	Mobile Terminated
NDC	National Destination Code
NMSI	National Mobile Subscriber Identity
PKI	Public Key Infrastructure

PLMN-ID	Public Land Mobile Network Identity
PMSI	Pseudo Mobile Subscriber Identifier
PS	Packet Switched
PSTN	Public Switched Telephone Network
RA	Routing Area
RAC	Routing Area Code
RAI	Routing Area Identity
RAN	Radio Access Network
RID	Recovery Identity
RNC	Radio Network Controller
SAT	SIM Application Toolkit
SGSN	Serving GPRS Support Node
S-GW	Serving Gateway
SIM	Subscriber Identity Module
SN	Serving Network
SS7	Signalling System 7
STK	SIM Toolkit
TA	Tracking Area
TAC	Tracking Area Code
TAI	Tracking Area Identity
TESLA	Timed Efficient Stream Loss-tolerant Authentication
TID	Transient Identity
TMSI	Temporary Mobile Subscriber Identity
UE	User Equipment
UICC	Universal Integrated Circuit Card
UMTS	Universal Mobile Telecommunications System
USAT	USIM Application Toolkit
USIM	Universal Subscriber Identity Module
USRP	Universal Software Radio Peripheral
USSD	Unstructured Supplementary Service Data
UTRAN	Universal Terrestrial Radio Access Network
VLR	Visitor Location Register

Chapter 1

Introduction

1.1 Context of Research

Mobile phones have become an integral part of our daily lives. We use them at work, at home and everywhere in between to stay connected to the world around us. The growth in mobile subscribers over the years has been very substantial. At present, approximately sixty-three percent of the world population are connected to a mobile network [121].

The major growth in use of mobile networks started with the introduction of the *second generation* (2G) GSM network in the early 1990s. We are now in the era of *fourth generation* (4G) mobile networks; however, legacy mobile systems, i.e. GSM and *third generation* (3G) mobile networks, continue to operate to support a large segment of the mobile subscribers [42, 134]. Even today, approximately forty-four percent of global connections use GSM technology and thirty percent use 3G mobile technology; as yet 4G mobile networks only support twenty percent of global connections [42]. Although 4G subscriptions are growing fast, legacy mobile systems remain of huge practical importance worldwide, and are unlikely to be replaced for many decades to come. Moreover, for practical reasons, all later generation mobile technology needs to support the earlier generation technology, which makes security flaws in legacy networks potential threats to all mobile subscribers. As a result, finding ways of improving the security offered by existing mobile systems is clearly of great practical significance.

A wide range of approaches [46, 48, 71, 81, 82, 108, 113, 116] have been proposed to improve the security and privacy features of GSM, 3G and 4G networks. However, they all require significant changes to the deployed network infrastructure; that is they require modifications to all the participating entities, i.e. to the serving networks, home networks, and mobile devices, as well as the network protocols. Unfortunately, it is clear

that making such modifications to widely deployed mobile systems is almost certainly impractical in practice. Prior to commencing the work described in this thesis, it was not known whether it is possible to improve the security and privacy properties of the current mobile systems without changing the deployed network infrastructure, i.e. the serving networks and mobile phones. Addressing this question has provided the main motivation for the work described in this thesis.

This chapter provides an overview of the thesis, and is organised as follows. In Section 1.2 we discuss the research motivation. Section 1.3 outlines the major contributions. A list of relevant publications is presented in Section 1.4, and Section 1.5 concludes the chapter with a brief outline of the thesis.

1.2 Motivation

While *first generation* (1G) mobile systems did not provide any security features, security has been an integral part of such systems since the advent of 2G. For example, GSM, perhaps the best known 2G system, provides a range of security features, including authentication of the mobile user to the network, data confidentiality across the air interface, and a degree of user pseudonymity through the use of temporary identities. However, the absence of network authentication in GSM leaves the system open to a wide range of threats (see, for example, [125, 130, 136]). Despite the introduction and deployment of 3G (UMTS) and 4G (LTE) mobile systems, which rectify this shortcoming by providing mutual authentication between phone and network, GSM remains of huge practical importance worldwide and is unlikely to be replaced for many decades to come. As a result, incorporating new features to improve the security offered by GSM, without the need for changes to deployed phones, access networks, and protocols, is of great practical importance. This observation motivates the investigation of a novel scheme to include network-to-phone authentication in GSM mobile systems in a way that is completely transparent to the deployed network infrastructure.

Although the security properties of the 3G and 4G mobile networks have significantly improved by comparison with GSM, significant shortcomings remain with respect to user privacy. The possibility of user tracking by intercepting air interface traffic was considered in the design phase of the second generation GSM network, and was addressed by the use of a changing pseudonym, the *temporary mobile subscriber identity* (TMSI), instead of the permanent *international mobile subscriber identity* (IMSI) for transmissions across the radio link. Newly allocated TMSIs are transmitted to the phone by the network in encrypted form to prevent linking of TMSIs or of a TMSI to the IMSI. The same mechanism was adopted in the 3G [162] and 4G [93] systems.

Although the temporary identity is used instead of the IMSI in most cases, in certain situations the IMSI is sent in cleartext by the mobile. One such case is when a mobile device is switched on and wishes to connect to a new network, and hence will not have an assigned temporary identity. Another case is where the network is unable to identify the IMSI from the presented temporary identity and is therefore obliged to request the transmission of the IMSI [31, 114]. An active adversary can exploit this and masquerade as a legitimate network to request the permanent identity [3]. Such an adversary is known as an 'IMSI catcher' [151], and is a threat to all generations of mobile networks. Although IMSI catchers were initially only available to government agencies because of their cost and complexity, advances in hardware technology and the availability of software has made IMSI catcher capabilities affordable for almost anyone. During the last couple of years, widespread use of unregulated IMSI catchers has been observed in a number of countries [45, 100, 150]. Security agencies have reported on the use of IMSI catchers by criminals and foreign intelligence agencies [66, 159], indicating that such attacks are now a serious issue.

Over the years, a wide range of possible solutions to the IMSI catcher problem have been proposed [48, 50, 71, 94, 108, 113, 142, 143]; unfortunately, they all require significant modifications to the air interface protocol, which would require changes to the operation of all the serving networks as well as all the deployed phones. As a result, it seems likely that making the necessary major modifications to the operation of the air interface after deployment is infeasible in practice. It would therefore be extremely valuable if a scheme to reduce the threat from IMSI catchers and thereby better protect user privacy could be devised which does not require significant changes to the existing network infrastructures and has minimal computational cost. This observation has motivated the work described in this thesis towards devising novel schemes to improve user identity privacy in mobile systems which could actually be deployed.

1.3 Contributions

In this thesis we describe the results of research into the possible enhancement of the security and privacy properties of the GSM, 3G and 4G mobile networks, as well as analyses of the deployability of certain previously proposed solutions. We further propose new schemes to add network authentication to GSM networks, and to enhance user identity privacy in the GSM, 3G, and 4G mobile networks without affecting the intermediate network entities, notably the serving networks and mobile equipment. Of the contributions, I formulated the research problem and came up with the proposed schemes, and my supervisor provided guidance and necessary corrections.

The main contributions of this thesis are as follows.

- Although many authors have proposed schemes designed to improve the security and privacy properties of GSM by providing mutual authentication and/or hiding the permanent subscriber identity, they all require changes to the existing radio interface protocols. It is somewhat counterintuitive to propose that authentication of the network to the phone can be achieved without modifying the way in which the existing access networks and phones operate. However, this is what we have done. This apparently paradoxical result is achieved by using a technique we refer to as RAND hijacking. This involves using the authentication 'challenge' *RAND*, which serves as a nonce in the existing unilateral authentication protocol and is sent from the network to the phone, to contain data which enables the recipient SIM to verify its origin and freshness. That is, the *RAND* is hijacked to act as a communications channel between a home network and a SIM. We propose a novel authentication scheme for GSM where, instead of generating the *RAND* at random, it contains information enabling the network to be authenticated; this information is sent in encrypted form so that to an eavesdropper it is indistinguishable from a random value. We provide a detailed analysis of the scheme and verify its correctness and security properties using ProVerif.

- We analyse certain recently proposed modifications to the operation of mobile systems intended to address user privacy threats. Specifically, we critically examine the proposals of Arapinis et al. [48]. This analysis reveals that the proposed modifications are impractical in a variety of ways; not only are there security and implementation issues, but the necessary changes to the operation of the system are very significant and much greater than was envisaged by the authors. In fact, some of the privacy issues appear almost impossible to address without a complete redesign of the security system, meaning that making significant system changes to address some of them are unlikely to be worth the effort. The shortcomings of the proposed 'fixes' exist despite the fact that the modifications have been verified using a logic-based modelling tool, suggesting that such tools need to be used with great care. We also suggest possible alternative approaches to some of the modifications.

- We address the decades-old privacy problem of disclosure of the permanent subscriber identity (IMSI), a problem arising in all generations of mobile networks that makes IMSI catchers a real threat. A number of possible modifications to existing protocols have been proposed to address the problem; however, almost

all require significant changes to existing deployed infrastructures, and are therefore impractical to implement in practice. We propose an approach which does not require any changes to the existing deployed network infrastructures, i.e. to the serving networks or mobile devices, but offers improved user identity protection over the air interface. The proposed schemes make use of multiple IMSIs for an individual subscriber, thereby offering a degree of pseudonymity for a user. The only changes required are to the operation of the authentication centre in the home network and to the SIM/USIM, both owned by a single entity. We present two different approaches to the use and management of multiple IMSIs, and report on experiments to validate their deployability. The schemes could be deployed immediately since they are completely transparent to the existing network infrastructure.

We further improve our proposed scheme to address a possible shortcoming. The improved scheme prevents disclosure of the subscriber's IMSI by using a dynamic pseudo-IMSI that is only identifiable by the USIM's home network. A major challenge in using dynamic pseudo-IMSIs is possible loss of identity synchronisation between a USIM and its home network, an issue which has not been adequately addressed in previous work. We present an approach for identity recovery to be used in the event of pseudo-IMSI desynchronisation. The scheme requires changes to the home network and the USIM, but not to the serving network, mobile phone or other internal network protocols, enabling simple, transparent and evolutionary migration. We discuss the strengths and limitations of the scheme, and verify its correctness and security properties using ProVerif.

1.4 Publications

Publications containing some of the research results described in this thesis are listed below.

- M. S. A. Khan and C. J. Mitchell, 'Another look at privacy threats in 3G mobile telephony', in: W. Susilo and Y. Mu (eds.), *Information Security and Privacy—19th Australasian Conference, ACISP 2014, Wollongong, NSW, Australia, July 7–9, 2014. Proceedings,* Springer-Verlag LNCS 8544, Berlin (2014), pp.386–396.

- M. S. A. Khan and C. J. Mitchell, 'Improving air interface user privacy in mobile telephony', in: L. Chen and S. Matsuo (eds.), *Security Standardisation Research, Second International Conference, SSR 2015, Tokyo, Japan, December*

15–16, 2015, Proceedings, Springer-Verlag LNCS 9497, Berlin (2015), pp.165–184.

- M. S. A. Khan and C. J. Mitchell, 'Retrofitting mutual authentication to GSM using RAND hijacking', in: G. Barthe, E. Markatos and P. Samarati (eds.), *Security and Trust Management—12th International Workshop, STM 2016, Heraklion, Crete, Greece, September 26–27, 2016, Proceedings,* Springer-Verlag LNCS 9871, Berlin (2016), pp.17–31.

- M. S. A. Khan and C. J. Mitchell, 'Trashing IMSI catchers in mobile networks', in: G. Noubir and M. Conti and S. K. Kasera (eds.), *10th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec 2017, Boston, MA, USA, July 18-20, 2017. Proceedings,* ACM (2017), pp.207–218.

All the previously mentioned publications were supervised by C. J. Mitchell. A further publication co-authored whilst conducting my PhD research is as follows.

- M. N. Kayuni, M. S. A. Khan, W. Li, C. J. Mitchell and P. Yau, 'Generating unlinkable IPv6 addresses', in: L. Chen and S. Matsuo (eds.), *Security Standardisation Research, Second International Conference, SSR 2015, Tokyo, Japan, December 15–16, 2015, Proceedings,* Springer-Verlag LNCS 9497, Berlin (2015), pp.185–199.

1.5 Thesis Outline

The reminder of this thesis is divided into four parts, as follows.

1. Part I describes necessary background material. It contains two chapters, as follows.

   - *Chapter 2* describes those aspects of the system architecture, user identities, air interface protocols, and services of 2G mobile systems necessary to understand the rest of the thesis.
   - *Chapter 3* outlines aspects of the system architecture, user identities, air interface protocols, and services of 3G and 4G mobile systems relevant to the rest of the thesis.

2. Part II describes the security and privacy threats arising from use of GSM. It further presents novel schemes to support mutual authentication, and to improve air interface user privacy in GSM networks. It contains the following three chapters.

- *Chapter 4* gives an overview of known security and privacy issues arising from the lack of mutual authentication and disclosure of the permanent subscriber identity in GSM networks. It also discusses previous attempts to address these issues.
- *Chapter 5* describes and analyses a novel scheme to provide mutual authentication in GSM without affecting the serving networks or phones.
- *Chapter 6* describes and analyses a scheme to improve air interface user privacy in GSM networks, that does not require modifications to the serving networks or phones.

3. Part III addresses privacy threats arising in 3G and 4G mobile systems. Novel schemes designed to address the threat of IMSI catchers in mobile networks are presented. It contains three chapters, as follows.

   - *Chapter 7* reviews known privacy issues arising from disclosure of the permanent subscriber identity in 3G and 4G. It also discusses previous attempts to address these issues.
   - *Chapter 8* presents a critical analysis of the proposed modifications to the operation of 3G mobile systems due to Arapinis et al. [48], intended to address threats to user identity privacy; it further proposes possible alternative means of mitigating these threats.
   - *Chapter 9* describes and analyses novel schemes to address the decades-old privacy problem of disclosure of the permanent subscriber identity, focussing in particular on 3G and 4G.

4. Part IV concludes the thesis by summarising the main contributions as well as highlighting possible areas for future work. This part of the thesis consists of a single chapter, *Chapter 10*.

Part I

Background

Overview

Part I describes necessary background material. It contains two chapters, as follows.

- *Chapter 2* describes those aspects of the system architecture, user identities, air interface protocols, and services of 2G systems necessary to understand the rest of the thesis.
- *Chapter 3* outlines aspects of the system architecture, user identities, air interface protocols, and services of 3G and 4G systems relevant to the rest of the thesis.

Chapter 2

2G Mobile Systems

2.1 Introduction

The 2G GSM[1] (*global system for mobile communications*) system, developed initially by the *European conference of postal and telecommunications administrations* (CEPT) and then adopted by the *European telecommunications standard institute* (ETSI), replaced the 1G analogue mobile systems. The first GSM system was developed in Finland by the operator Radiolinja in 1991, and GSM remains the most widely-used technology for wireless networks worldwide [42]. This chapter introduces those parts of the GSM technology relevant to the remainder of the thesis, with a particular focus on the security and privacy features.

The chapter is organised as follows. In Section 2.2 we briefly describe the architecture of GSM. Section 2.3 outlines the identities used within a GSM network. In Section 2.4 we describe in detail the GSM authentication protocol, which underlies all the GSM security and privacy features. Section 2.5 outlines the GSM services key to this thesis. In Section 2.6 we describe relevant features of a SIM. Section 2.7 concludes the chapter by briefly reviewing the *general packet radio service* (GPRS), a GSM-based 2G data communications technology.

2.2 System Architecture

Although the only visible part of a mobile network architecture is the set of interconnected and stationary radio towers distributed over the geographical coverage area of a *mobile network operator* (MNO), there is a complex network behind the scenes providing seamless wireless communication services. In this section we give a simplified

[1]Originally known as groupe spècial mobile, GSM was renamed following the worldwide adoption of the standard.

overview of the architecture of the GSM network, focusing on those entities that are of relevance to this thesis. Figure 2.1 illustrates the architecture and the connections between these components. Further details can be found in technical specifications GSM 03.02 [87] and 3GPP TS 43.020 [40].

2.2.1 Overview

The infrastructure for a GSM network can be divided into two subsystems: the *base station subsystem* (BSS) and the *network subsystem*. The stationary radio towers belong to the BSS. To attach to a network, a mobile phone, a *mobile station* in GSM terminology, connects via its radio interface to a radio tower. The radio tower, also known as a *base transceiver station* (BTS) or simply a *base station* (BS), relays radio traffic to and from the mobile network and provides over-the-air access to the network. Each BTS covers a specific area called a *cell*, and is identified by its cell identity. Cells are grouped together to form a *location area* (LA), which is controlled by a single *base station controller* (BSC). The BTSs and BSCs form the BSS, also known as the *radio network*, which contains the functionality necessary to enable mobile users to connect to the network over the radio interface. The radio interface is usually referred to as the *air interface*.

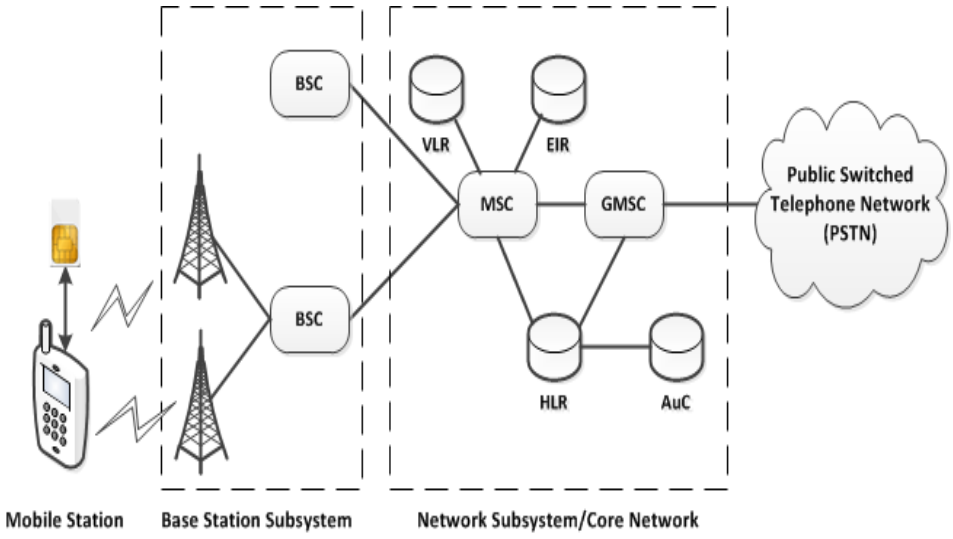

Figure 2.1: GSM system architecture (simplified)

The network subsystem is the back-end component of the GSM network architecture, and is known as the *core network*. The *mobile switching center* (MSC) is the controlling element of a core network. A number of BSCs are connected to a single MSC, and the MSC is responsible for controlling call setup and routing, and for

mobility management. The *gateway-MSC* (GMSC) connects the mobile network to the *public switched telephone network* (PSTN)—the global telephony network. Two separate databases contain subscriber account information, namely the *home location register* (HLR) and the *visitor location register* (VLR); these databases form part of the core network, and are connected to the MSC to support its operation, for example, for authentication of subscribers, location management, and organizing handover to a BSC. The VLR holds dynamic information regarding the subscribers currently roaming in the jurisdiction of the MSC, information which is maintained until the subscriber is handed over to a different MSC.

Each MNO maintains a HLR that is used to store information about its subscribers. The HLR holds a record for each subscriber, where a record contains the following information:

- the subscriber's permanent identity, known as the IMSI;
- the subscriber's phone number;
- the set of services available to this subscriber;
- the identity of the MSC currently responsible for forwarding mobile terminated services to this subscriber; and
- the subscriber-specific authentication data.

Although a subscriber is normally assigned only a single phone number, an HLR could store multiple phone numbers for a subscriber. Generally there is only one HLR for an MNO; however, an MNO could implement the HLR in a distributed way to support a large numbers of users. The HLR is associated with an *authentication center* (AuC) that stores the cryptographic credentials required for communicating with the SIM; specifically, the AuC shares a unique secret key K with each SIM issued by the MNO to which it belongs, and is responsible for computing subscriber-specific authentication data.

The *equipment identity register* (EIR) is a database of the identities of mobile devices used in the global GSM network. A mobile device identity, known as *international mobile equipment identity* (IMEI) (see Section 2.3.2 below), can be classified as *white-listed*, *grey-listed*, or *black-listed*. The EIR enables the MSC to prevent specific mobile devices, e.g. banned or stolen mobile phones, from accessing its services. The possible use of the EIR in a GSM network is discussed in Section 4 of the technical specifications GSM 02.16 [85] and 3GPP TS 22.016 [21]. Each MSC of a GSM network is connected to the EIR; a centralised EIR could serve the MNOs in a specific region.

2.2.2 Security Context Classification

The entities in GSM network architecture can be grouped into the three categories described below, where the categories are based on the trust inter-relationships. We use these categories when describing security-related functionality in the remainder of the thesis.

2.2.2.1 Mobile Station

A complete mobile phone is referred to as a *mobile station* (MS), where the term encapsulates not only the *mobile equipment* (ME), i.e. the phone, but also the *subscriber identity module* (SIM) within it, where the SIM takes the form of a cut-down smart card. The ME is made up of components which may have been designed and manufactured by a range of vendors, whereas the SIM is always owned and managed by the subscriber's MNO. The ME must support all the mandatory features of the GSM standards, and is known as a *2G ME*. The SIM embodies the relationship between the human user and the issuing MNO, including the IMSI, the mobile number, and other user (subscriber) data, together with a secret key K shared with the issuing network which forms the basis for all the air interface security features. Further details of the SIM are given in Section 2.6.

2.2.2.2 Serving Network

The radio network, along with the MSC and its associated VLR, forms part of the *serving network* (SN). This is the network from which a subscriber receives telecommunication services. To support roaming subscribers, the SN could be managed by an MNO different from the MNO with which the subscriber has a contract for the provision of mobile services. This arrangement needs to be supported by a roaming agreement between the MNOs concerned. Given the absence of a subscriber agreement between a roaming subscriber and an SN, there is no direct trust relationship between them, although there is an indirect relationship since the SN is trusted by the home network, as described in the next section. However, the SN must follow the protocol specification when communicating with the MS and home network.

2.2.2.3 Home Network

The HLR and the AuC operated by a single MNO together form the *home network* (HN) component of a mobile network. The HN is the source of trust for the user of a mobile network, and is individually managed by each MNO. The HN is responsible for subscriber authentication, authorisation, and cryptographic key generation. As

mentioned above, the AuC stores the subscriber's secret key K, and is responsible for computing authentication data and cryptographic keys for an individual subscriber; given its sensitivity, access to the AuC is restricted to the HLR.

2.3 System Identities

2.3.1 IMSI

An IMSI is a string of 15 decimal digits which uniquely identifies a GSM subscriber worldwide; it is used for subscriber-related signalling in the network. The structure of an IMSI is defined in technical specifications GSM 03.03 [5], 3GPP TS 23.003 [23] and E.212 [157]. As stated above, it is stored in the subscriber's SIM and in the HLR; it is the key to all information about subscribers. An IMSI is made up of the following three parts (see Figure 2.2).

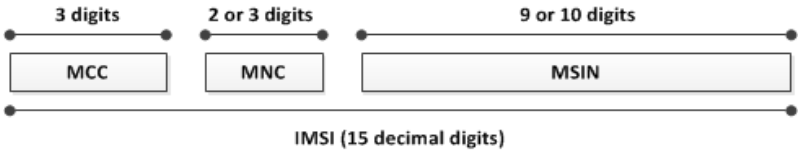

Figure 2.2: Structure of an IMSI

- *Mobile Country Code* (MCC): The first three digits of an IMSI are the MCC that identifies the subscriber's home country. The allocation of MCCs is administered by the ITU-T (the Telecommunication standardisation sector of the International Telecommunications Union)[2].

- *Mobile Network Code* (MNC): The next two or three digits uniquely identify the MNO of a subscriber within the country identified by the MCC, and are known as the MNC. The length of the MNC, i.e. whether it is two or three digits, is a national matter, and MNC values are managed by the national administrator. A unique MNC is necessary because there are usually multiple mobile networks within a country. In the United Kingdom, for example, the following MNCs are used: 10 for O2, 15 for Vodafone, 30 for T-Mobile, and 33 for Orange.

- *Mobile Subscriber Identification Number* (MSIN): The remaining nine or ten digits form the MSIN, and are administered by the network operator. The MSIN uniquely identifies a subscriber of an MNO.

[2] `http://www.itu.int/en/ITU-T/Pages/default.aspx`

IMSIs thus have geographical significance, and are typically managed by the network operator in blocks. The MCC and MNC combination uniquely identifies the HN, and is known as the *public land mobile network identity* (PLMN-ID); analogously, the MNC and MSIN combination uniquely identifies a subscriber within a national context, and is known as the *national mobile subscriber identity* (NMSI). The MSIN is used by the network operator to identify the subscriber for billing and other operational purposes. So, each IMSI uniquely identifies the mobile user, as well as the user's HN and home country; hence, it has privacy and security significance. The IMSI for a subscriber is normally fixed.

2.3.2 IMEI

An IMEI is a string of 15 decimal digits used to uniquely and permanently identify an ME, allowing a stolen equipment to be blacklisted in the EIR. It plays no role in the provision of communication services, and a network cannot verify its authenticity. It might be accompanied by a *software version* number, in which case the identity is called an *international mobile equipment identity software version* (IMEISV). Because of possible software upgrades to a terminal, the IMEISV might change during the terminal's lifetime, while the IMEI remains the same. Although the technical specifications [6, 24] describe an IMEI as temporary subscriber data, subscribers typically use a mobile device for a significant period of time; hence, an IMEI is quasi-permanent, and if sent in cleartext it could compromise user privacy. Further details can be found in technical specifications GSM 03.03 [5], and 3GPP TS 23.003 [23].

2.3.3 MSISDN

The phone number of a subscriber, known in GSM as the *mobile station international ISDN number* (MSISDN) [86], is a string of up to 15 decimal digits, of which the first three are the *country code* (CC), the international code of the subscriber's home country. The remaining digits are the national mobile number, which consists of a *national destination code* (NDC) and a unique *subscriber number*. The composition of the MSISDN is such that it can be used as a global address in the core network for routing data to the HLR of the MS. The CC and the NDC provide the necessary routing information, and the MSISDN is used in delivering *mobile terminated* (MT) services (see Section 2.5.4). If further routing information is required, it should be contained in the first few digits of the subscriber number [5, 158]. Since the introduction of *mobile number portability* blurs the significance of the NDC, routing data can if necessary be obtained by other means. Further details can be found in the relevant

standards [5, 23, 158].

The MSISDN is typically public information for a subscriber. It is not used to identify a subscriber in the network, and is not included in the signalling messages sent across the air interface. As a result, it is outside the scope of this thesis, and we do not consider it further here.

2.3.4 TMSI

As discussed in Section 1.2, a TMSI is a temporary subscriber identity used in place of an IMSI in transmissions across the air interface, with the goal of providing user pseudonymity. The TMSI is allocated by the MSC, and is transferred to the MS via an encrypted channel. The MSC maintains the relationship between a TMSI and its associated IMSI. Since the MSC uses the subscriber's IMSI in communications across the core network, maintaining the mapping of an IMSI from a TMSI is critical for successful network operation. We discuss this issue in detail in the next chapter (see Section 3.7).

A TMSI consists of four octets. Since a TMSI has only local significance, i.e. within an MSC and the area controlled by an MSC, its structure and coding are chosen by the SN [5, 23]. When a subscriber moves from one LA to another, its TMSI is updated by the SN. Although a subscriber is temporarily traceable via its TMSI, the length of time a single TMSI remains valid is limited, and is configurable by the SN.

2.3.5 LAI

A *location area identity* (LAI) uniquely identifies an LA within a mobile network, where LAs are geographical sub-divisions of the area covered by a single MSC. An LAI is a combination of the PLMN-ID of the mobile network and a *location area code* (LAC). A LAC is two bytes long, and is managed by an MSC [5, 23].

2.4 Authentication Protocol

To prevent unauthorised MSs gaining access to network service, GSM incorporates an authentication procedure which enables the network to verify that the SIM in an MS is genuine, and has the identity it claims. This procedure is known as the GSM *authentication and key agreement* (AKA) protocol; this protocol is at the core of air interface security, since it also establishes a session key used for subsequent traffic encryption.

The AKA is performed between the MSC of the SN and the MS as part of a range of network operations, including when:

- a change is necessary to the subscriber-related information element in the VLR or HLR, including one or both of;
  - a location update involving change of MSC,
  - the activation or deactivation of a supplementary service;
- an MS attempts to access a service, such as a mobile originating or a mobile terminating service; or
- an MS makes its first network access after a restart.

Although user identification is not part of the AKA protocol, it is an implicit prerequisite for executing the protocol. In the user identification phase, the MS will, whenever it can, identify itself using its TMSI, and the MSC determines the IMSI from the TMSI using its own process. If the MSC fails to determine the IMSI from the supplied TMSI, it requests the MS to send its IMSI using a *user identity request* message; the MS responds with its cleartext IMSI in a *user identity response* message.

Once the SN learns the IMSI, it selects the next unused element from an IMSI-specific list of *authentication vectors* (AVs), stored in the SN's VLR. The AV contains all the data necessary to perform AKA. If the list is empty, the SN determines the subscriber's HN by parsing the IMSI, and contacts this network to request a new set of AVs, which are typically sent in small batches (to overcome latency).

On receipt of such a request, the HN's AuC retrieves the secret key K for the specified IMSI, and uses it to generate a batch of AVs. These AVs are generated using a pair of cryptographic functions known as $A3$ and $A8$. These functions only need to be implemented by the HN and the SIMs it issues, and the choice of functions is up to the network; nevertheless, the *3rd generation partnership project* (3GPP)[3] recommends use of the corresponding 3G functions in a modified way [41]. The function $A3$ is a 32-bit MAC generating function and $A8$ is a 64-bit *key derivation function* (KDF).

Generating an AV involves the following steps (see Figure 2.3(a)); which can be repeated as necessary to generate a batch.

1. A 128-bit random (or pseudorandom) value $RAND$ is generated.
2. The AuC computes the 32-bit value $XRES$ as $XRES = A3_K(RAND)$.

[3] `http://www.3gpp.org`

3. The AuC computes the 64-bit key K_c as $K_c = A8_K(RAND)$.

4. $AV = (RAND, XRES, K_c)$.

The generated AVs are then sent back to the SN, which can then use one of them to conduct AKA.

The core of AKA is a challenge-response protocol. The messages exchanged among the involved parties, i.e. the MS, the SN, and the HN, are shown in Figure 2.4, and the computations involved are shown in Figure 2.3.

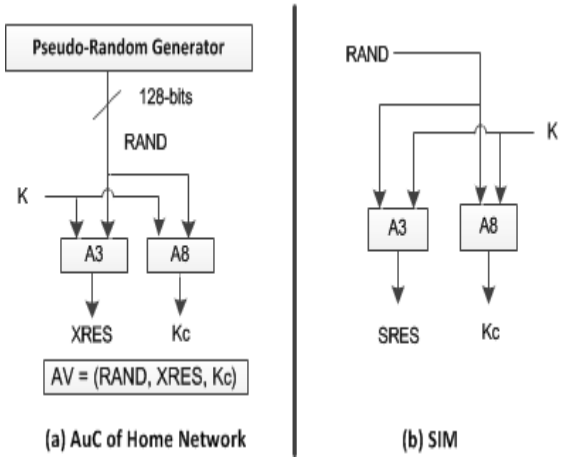

Figure 2.3: Computations of GSM AKA key values

The challenge-response procedure operates as follows. Further details can be found in technical specifications GSM 03.20 [2], 3GPP TS 43.020 [40] and GSM 04.08 [7].

1. The serving MSC sends *RAND* to the MS as an authentication challenge. The ME passes the received *RAND* to the SIM using the *RUN GSM ALGORITHM* command.

2. The SIM computes $SRES = A3_K(RAND)$ and $K_c = A8_K(RAND)$, where $A3$ and $A8$ are the same functions as used by the HN to generate the AV, *SRES* is a 32-bit *signed response*, and K_c is a 64-bit session key used to encrypt data sent across the air interface. Note that precisely the same computation was performed by the AuC to generate *XRES* and K_c earlier (see Figure 2.3(b)).

3. The SIM returns *SRES* and K_c to the ME. The ME keeps the session key K_c to use in data encryption, and forwards *SRES* to the serving MSC.

4. The serving MSC compares the received *SRES* with the value of *XRES* from the AV; if they agree then the MS is deemed authenticated.

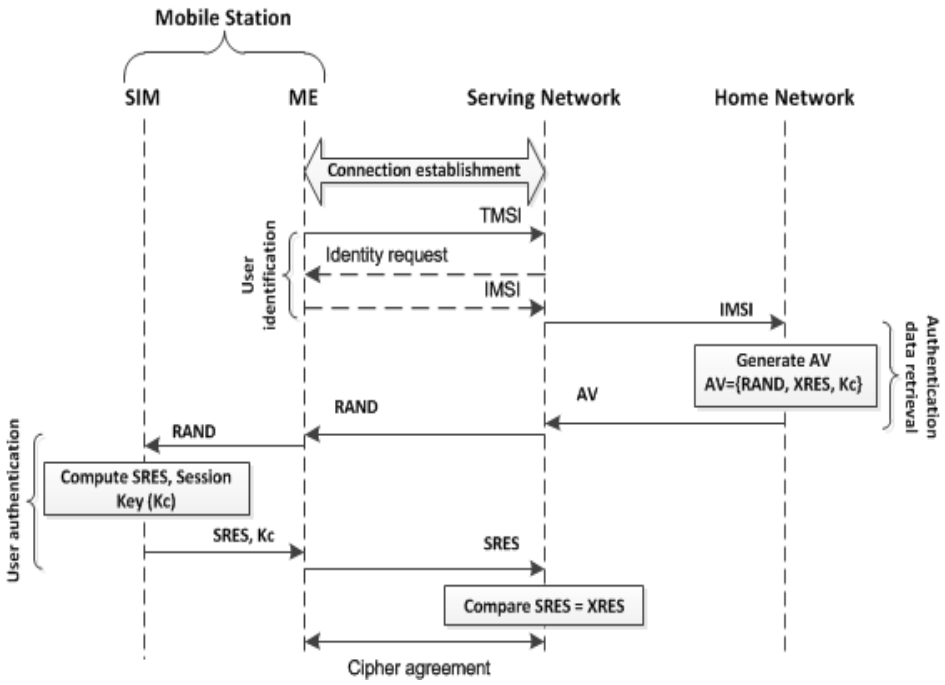

Figure 2.4: GSM AKA message flow

After successful authentication of an MS by a serving MSC, both the MS and the MSC hold a valid session key K_c. The MSC transfers this key to the appropriate BSC, where it is used for traffic encryption using one of the standardised algorithms (i.e. one of $A5/1$, $A5/2$ and $A5/3$), as selected by the BSC of the SN. The serving MSC also assigns a TMSI and sends it to the MS; the BSC will encrypt the TMSI prior to transmission across the air interface, just as it does for voice traffic.

2.5 Network Activities

We next summarise certain network operations of particular relevance to security and privacy.

2.5.1 IMSI Attach

After a mobile device is switched on, its first action is to register with the network to enable it to send and receive calls. This is achieved using the IMSI attach process.

Radio communication between the MS and the SN takes place over various types of channel, including dedicated channels, shared channels and broadcast channels. When a mobile device is switched on, it listens for a *SYSTEM INFORMATION* message advertised by a network over a broadcast channel known as the *broadcast common control*

channel (BCCH). Such messages contain a range of information about the network, including the PLMN-ID of the radio tower; the identification of the radio tower, consisting of the LAC and the cell identity; and the frequencies used by neighbouring cells. On receiving the message, the MS compares the PLMN-ID values in the message with the list of permitted networks stored in its SIM. If there is a match, the MS responds to the network with a *channel request* and, as soon as a channel is assigned, the MS initiates the attachment process (see Figure 2.5).

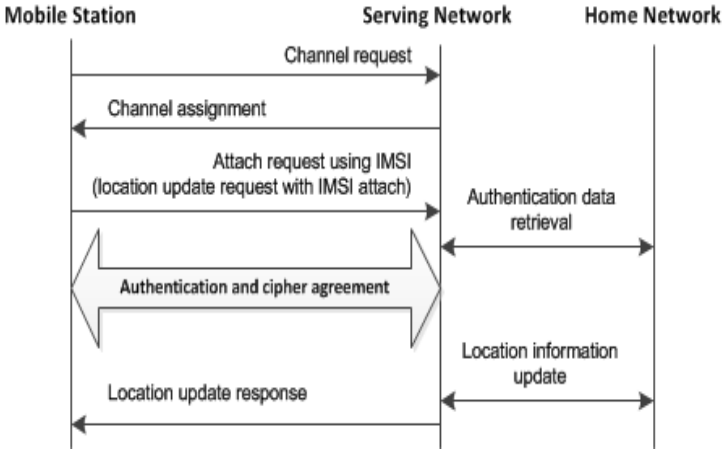

Figure 2.5: Steps in mobile station attachment

The IMSI attach procedure is described in Section 4.4.3 of technical specification GSM 04.08 [7]. It is accomplished using the location update procedure, where the *update type* information element in the *location update request* message sent by the MS to the network indicates IMSI attach. This *location update request* message includes the IMSI of the MS. On receiving the message, the SN authenticates the MS using AKA, described in Section 2.4. If authentication is successful, the SN updates the location information of the subscriber at the subscriber's HN (see Section 2.5.2 below), and notifies the MS; otherwise, the SN sends the MS a reject message. After a successful IMSI attach, the MS is sent a TMSI, which is used as its identifier in subsequent communication.

2.5.2 Location Update

When a subscriber moves to a new location area, the HN must be notified so that the MS can continue to receive mobile-terminated services (see Section 2.5.4), e.g. receiving calls or texts. An MS can initiate a location update procedure with the SN for a range of reasons, e.g. for a periodic location update to inform the network of its presence in a certain location area; normal location update because of a change of location

area; or during an IMSI attach. The reason for the location update is indicated in the *update type* field in the *location update request* message [7]. The main steps in the location update procedure are listed below. Further details can be found in technical specifications GSM 03.20 [2], GSM 04.08 [7], and GSM 03.12 [83].

1. The MS sends a *location update request* message containing its TMSI and LAI to the MSC of the SN.

2. If the MSC in the SN is the same as that indicated by the LAI, the serving MSC determines the IMSI from the TMSI using its VLR; otherwise, it follows the process described in technical specification GSM 09.02 [8] to collect the corresponding IMSI, along with any unused AVs, from the 'old' MSC, i.e. the MSC which was most recently connected to the MS. If the serving MSC fails to determine the IMSI by this method, it requests the MS to send its IMSI using a *user identity request* message, and the MS responds with its cleartext IMSI in a *user identity response* message.

3. The SN performs the AKA protocol, enables ciphering with the new key, and allocates a new TMSI to the MS as described in Section 2.4; it then initiates a location update with the HN, involving the following steps.

   (a) The serving MSC sends its identifier and the IMSI in a *location update request* message to the subscriber's HN.

   (b) On receiving the message, the HN updates the location information for the subscriber identified by the IMSI, and sends a *cancel location request* to the 'old' MSC for this subscriber.

   (c) The old MSC sends an acknowledgement for the location cancellation to the HN.

   (d) The HN sends an acknowledgement for the location update request to the initiating MSC.

4. The SN sends a *location update accept* or *location update reject* message to the MS.

2.5.3 Paging

Paging is used by an SN to locate an MS to which a connection needs to be established. In order to deliver a service to a user, the serving MSC needs to know the precise location of the MS. An ME typically reverts to an idle state most of the time to save

stored battery energy, and is therefore not in constant contact with the SN; thus the SN does not know which radio tower provides the best radio signal to the MS. However, the SN is aware of the location area in which the MS was most recently, and so it broadcasts a *paging* message throughout the location area to alert the MS.

A paging message is sent from the SN to all mobile devices in a particular area, and contains either an IMSI or a TMSI. The GSM standard [7] specifies three types of paging requests, known as *type 1*, *type 2*, and *type 3* paging messages. The three paging message types allow different numbers of MSs to be addressed with a single message. Types 1, 2 and 3 paging messages respectively allow one or two, two or three, or four MSs to be paged simultaneously. If an MS detects a paging message containing its IMSI or its current TMSI, it establishes a dedicated channel with the SN and sends the network a *paging response* message containing its current TMSI.

2.5.4 Mobile Terminated Services

How a network service is managed varies depending on whether the service originates or terminates at an MS. A *mobile terminated* (MT) service is one that is passively received by a subscriber, e.g. receiving a phone call or a short message.

The MT services are supported by the *paging* procedure. When an MT service is to be delivered, the core network first determines the responsible MSC for the target subscriber with the help of the subscriber's HLR. Next, the MSC obtains the location information for the destination subscriber from the VLR, and sends a *paging message* to all BSCs in the subscriber's location area. The message contains the identity of the subscriber, usually either an IMSI or a TMSI. The BSC broadcasts the *paging message* via all the BTSs in its jurisdiction. If the target subscriber responds with a *paging response* message, it undergoes an authentication, ciphering and service setup procedure with the SN; the pending service is subsequently delivered to the MS. Paging in mobile systems could be spoofed; we discuss this issue further in Section 4.5.5.

2.6 The SIM

The SIM is the cornerstone of GSM security, since it contains the IMSI and the associated 128-bit permanent key K. The cryptographic mechanisms used in authentication and key generation, i.e. $A3$ and $A8$, are implemented in the SIM. The properties of the SIM are specified in technical specifications GSM 02.17 [84] and 3GPP TS 42.017 [4]. In the original GSM specification and until release 4 of the 3GPP specifications[4], the physical smart card itself was called a SIM, so the specifications use the term 'a SIM

[4] `http://www.3gpp.org/specifications/specification-numbering`

card'. From release 4 of the 3GPP specifications, the smart card itself is called a *universal integrated circuit card* (UICC), and the term SIM now refers to an application running in the UICC. This change of approach allows for later generation network access applications, e.g. the *universal subscriber identity module* (USIM) and the *IMS SIM* (ISIM), to run on the same smart card platform. The UICC supports communications between the ME and an application in a UICC through well defined *commands*. The properties of the UICC are defined in technical specification ETSI TS 102.221 [88].

In this section we briefly describe the data storage capabilities of a UICC, focussing on the SIM and the USIM application, communications between an application residing in a UICC and the ME, and a feature of the SIM, known as the *SIM application toolkit*, that is of great use in designing the schemes we describe later in the thesis.

2.6.1 Memory Structure

A UICC stores data in a file system organised as a rooted tree with at most one file associated with each node. File types included *elementary files* (EFs) and *dedicated files* (DFs). Each EF stores a set of data units, records or objects. EFs are always leaf nodes in the tree, and DFs are used to group EFs; thus a DF is only a leaf node if it is empty. The *master file* (MF) is a mandatory and unique DF that serves as the root of the file system tree. Files can be addressed in various ways, depending on the file type; for example, they can be addressed by variable-length names or by two-byte identifiers, which may be concatenated to absolute and relative paths. Basic actions, e.g. read or update, on UICC data are controlled by access conditions which must be satisfied before the action can be performed. The access control policies are defined in 3GPP specifications [10, 12].

An application in a UICC consists of a set of functions, and has an associated subtree in the file system. The application's root DF is called an *application dedicated file* (ADF). Such ADFs are addressed by their unique DF name which must be listed in the EF_{DIR} — an application-independent file that is a descendant of the MF. When an ME initially accesses the UICC, if the EF_{DIR} is not found in the UICC, or if the USIM application, which has identifier ADF_{USIM}, is not listed in the EF_{DIR}, the UICC selects the SIM application; hence, a UICC must always contain the SIM application.

Both the SIM and USIM applications store a range of data about the subscriber, the supported services, and the network. For example, they store the subscriber's IMSI, phone number, current location, and subscriber-specific cryptographic credentials. These data are stored in specific EFs, e.g. the IMSI is stored in the EF_{IMSI}. The SIM data storage requirements are specified in Section 6 of technical specifications GSM 02.17 [84] and 3GPP TS 42.017 [4]. Figure 2.6 shows the simplified file structure

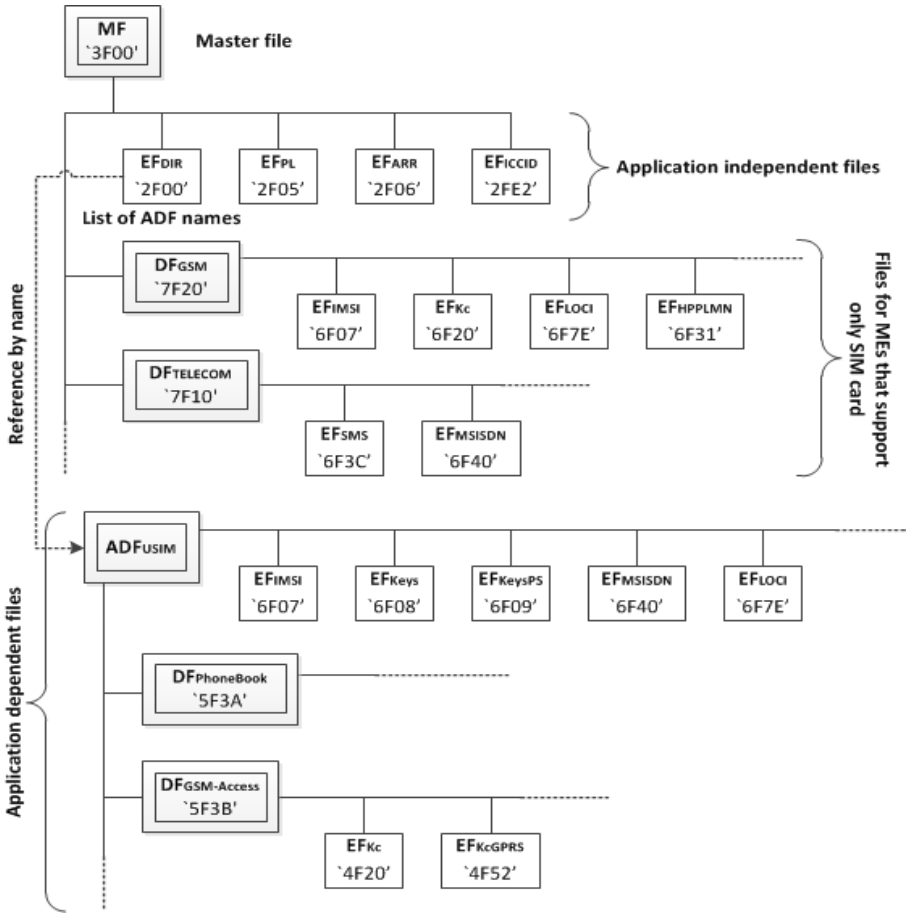

Figure 2.6: UICC file structure (simplified) [12, 28]

of a UICC supporting both the SIM and the USIM applications.

Although the EF_{IMSI} and the EF_{MSISDN} files are present for both the SIM and the USIM applications, their file names and the file identifiers for the two applications are identical (see Figure 2.6). These *EFs* contain the same information for use in different networks. This allows for memory efficient implementation of a SIM together with a USIM, since the files can be shared by the two applications.

2.6.2 Application Protocol Data Units

Data is exchanged between an application residing in a UICC and the ME in the form of *application protocol data units* (APDUs). An APDU sent by the ME is called a *command APDU*, and is answered by the application residing in the UICC using a *response APDU*. A matching pair of messages is referred to as a *command–response* pair.

Figure 2.7(a) shows the structure of a command APDU. The first four bytes, known as the *command header*, are mandatory. The *class* byte specifies a group of commands, i.e. the value 'A0' indicates commands for the SIM application. The code transmitted in the *instruction* byte indicates a specific command, telling the application which operations to perform. Command arguments are encoded in the parameter bytes *P1* and *P2*. The remaining parts of the command APDU are optional, where the *command data length* denotes the number of input bytes contained in the *command data* field, and is present if the command APDU carries additional data. The *response data length* encodes an upper bound for the size of the result data in the expected response APDU.

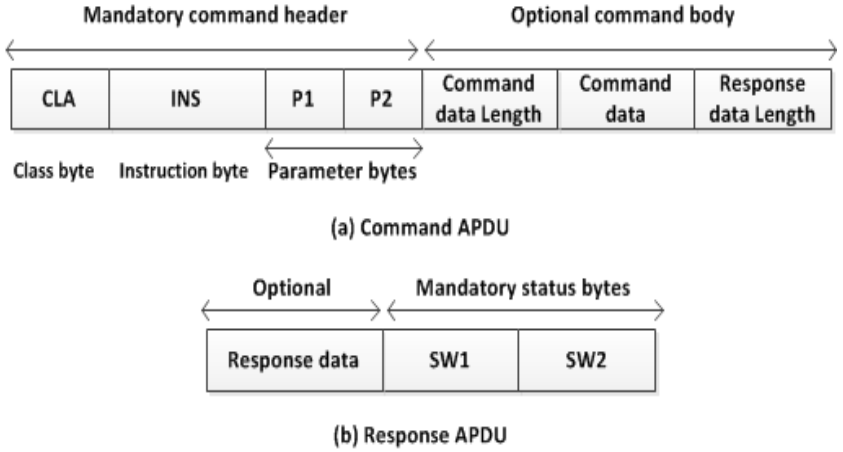

Figure 2.7: Command and response APDU

Figure 2.7(b) shows the structure of a response APDU. The *response data* is optional, and is present if the instruction specified in the command APDU generates response data. The *status bytes*, SW1 and SW2, are mandatory, and respectively categorise the outcome of the finished operation, and inform the ME about necessary follow-up commands. Further details can be found in technical specification ETSI TS 102.221 [88].

2.6.3 The SIM Application Toolkit

The *SIM application toolkit* (SAT) is a service operating across the SIM–ME interface that provides a mechanism for a SIM to initiate an action to be taken by the ME. As discussed in the previous section, communications between an ME and a SIM are command–response based; the SAT provides a set of commands, also known as *SIM toolkit* (STK) commands, which allow a SIM to initiate an action with an ME. A SIM that supports the SAT is known as a *proactive SIM*.

The GSM technical specification [12] states that an ME must communicate with a SIM using either the T=0 or T=1 protocol, specified in ISO/IEC 7816-3 [106]. In both cases the ME is always the master and thus initiates commands to the SIM; as a result there is no mechanism for the SIM to initiate communications with the ME. This limits the possibility of introducing new SIM features requiring the support of the ME, as the ME needs to know in advance what actions it should take. The proactive SIM provides a mechanism that allows the SIM to indicate to the ME, using a response to an ME-issued command, that it has some information to send. The SIM achieves this by including a special status byte '91' followed by the length of the instruction to send in the response APDU, which signals both the successful termination of the ME's last command as well as the length of the information describing the pending proactive command. The ME is then required to issue the *FETCH* command to retrieve the proactive command [9, 13]. The SIM responds with the proactive command, and the ME must execute it and return the result in the *TERMINAL RESPONSE* command. To avoid cross-phase compatibility problems, this service is only permitted to be used between a SIM and an ME that support the STK commands. The fact that an ME supports specific STK commands is revealed when the ME sends the *TERMINAL PROFILE* command during SIM initialisation.

The SIM can make a range of requests using the SAT service. Examples include: requesting the ME to display SIM-provided text, initiating the establishment of on-demand channels, and providing local information from the ME to the SIM. Although support of STK commands is optional for an ME, if an ME claims compliance with a specific GSM release then it is mandatory for the ME to support all functions of that release [13]. Since 1998 almost all of the MEs produced have been SAT-enabled, and today almost every ME on the market supports SAT. Further details can be found in technical specifications GSM 11.14 [13] and 3GPP TS 51.014 [9].

2.7 General Packet Radio Service

The *general packet radio service* (GPRS) is a GSM-based 2G data communication technology. Originally, GSM only supported *circuit-switched* (CS) data, for voice communications and short messages. GPRS is a backward-compatible update to GSM supporting the transfer of *packet-switched* (PS) data in a GSM network. Support for PS data is achieved by including additional entities, i.e. the SGSN and the GGSN, in the core network.

The GPRS security functions, i.e. the authentication of a subscriber and cipher key management, remain the same as in GSM. However, there are changes to certain

network operations. Significant changes in GPRS systems include the following.

- GPRS introduces an additional temporary subscriber identity, known as the *P-TMSI*.
- GPRS uses a separate session key for GPRS data encryption.
- GPRS employs a new location identity, known as the *routing area identity* (RAI), analogous to the LAI in GSM.

Chapter 3

3G and 4G Mobile Systems

3.1 Introduction

Third generation (3G) mobile systems, such as the *universal mobile telecommunications system* (UMTS), are the successor to 2G mobile technology, and represent an evolution of the GSM mobile networks. Analogously, fourth generation (4G) mobile systems, represented by the *long-term evolution* (LTE) system, introduce an all-*Internet protocol* (IP) based network infrastructure, as part of their evolution of 3G mobile technology. In this chapter we briefly describe the 3G and 4G mobile systems, explaining the relevant features and providing the terminology we use throughout the thesis. We describe in detail the authentication schemes used in these systems, which form a key part of the research results described.

The remainder of the chapter is structured as follows. In Section 3.2 we briefly describe the architecture of 3G and 4G mobile systems. Section 3.3 outlines the identities in 3G and 4G networks relevant to this thesis. In Section 3.4 we describe in detail the 3G and 4G authentication protocols. Section 3.5 briefly describes the relevant features of a USIM. In Section 3.6 we re-examine the network activities described in Section 2.5 in the context of 3G and 4G networks. Section 3.7 concludes the chapter by briefly reviewing how a subscriber's temporary identities are managed in mobile systems, a topic of particular relevance to the schemes we describe later in this thesis.

3.2 System Architecture

In this section we present a simplified description of the network architecture for 3G and 4G mobile systems. A detailed description of the 3G network architecture can be found in technical specification 3GPP TS 23.101 [16], and the 4G network architecture

is described in detail in technical specifications 3GPP TS 23.401 [25] and 3GPP TS 36.401 [39].

3.2.1 3G Mobile Systems

Like GSM mobile systems, the network infrastructure of a 3G network can be divided into two main parts, namely the *radio access network* (RAN) and the *core network*. There are two types of RAN in a 3G system. The *universal terrestrial radio access network* (UTRAN) is the newly introduced 3G RAN, whereas the *GSM/EDGE radio access network* (GERAN) is the GSM radio access network. The stationary radio towers provide the network-based termination point for the radio interface, and are known as *Node B* in UTRAN and BTS in GERAN. Multiple radio towers can be connected to the controlling unit of the RAN, referred as the *radio network controller (RNC)*. To attach to a 3G network, a mobile phone, a *user equipment* (UE) in 3GPP terminology, connects via its radio interface to a RAN, which is itself connected to the core network.

The services provided by the core network are divided into two domains: the *circuit-switched* (CS) domain, and the *packet-switched* (PS) domain. The network elements in the CS domain are the MSC, its associated VLR, and the GMSC; the functions of these entities are the same as those in a GSM network (see Section 2.2.1). The *serving GPRS support node* (SGSN) and the *gateway GPRS support node* (GGSN) are the entities in the PS domain. PS connections are managed by the SGSN, where the SGSN maintains its own VLR to store data about these connections. The roles of this VLR are the same as those of the MSC's VLR. Analogously, the SGSN is the counterpart of the MSC for the PS domain. The GGSN is responsible for controlling IP services both within the MNO and to the outside world, such as the Internet.

The MSC, GMSC, SGSN, and GGSN are connected to the MNO's HLR. Analogously to GSM, the HLR is associated with an AuC. The functions of the HLR and the AuC in 3G systems are similar to their GSM counterparts. However, there are differences in the detailed operation of the 3G HLR and AuC compared to GSM, necessary to support the 3G authentication protocol (see Section 3.4). Figure 3.1 gives a simplified view of the 3G systems architecture and of the connections among its components.

The UE encapsulates both the ME and the USIM, where a USIM is an application residing in a UICC. The ME contains the radio functionality and the protocols required to communicate with 3G network, i.e. it supports both the 2G and 3G RAN; it is also known as a 3G ME. The USIM consists of a group of functions for network access, and stores a wide range of data about the subscriber and the network, including the subscriber's IMSI. Further details of the USIM are given in Section 3.5.

The SN is responsible for communications with the UE via its RAN; hence, the

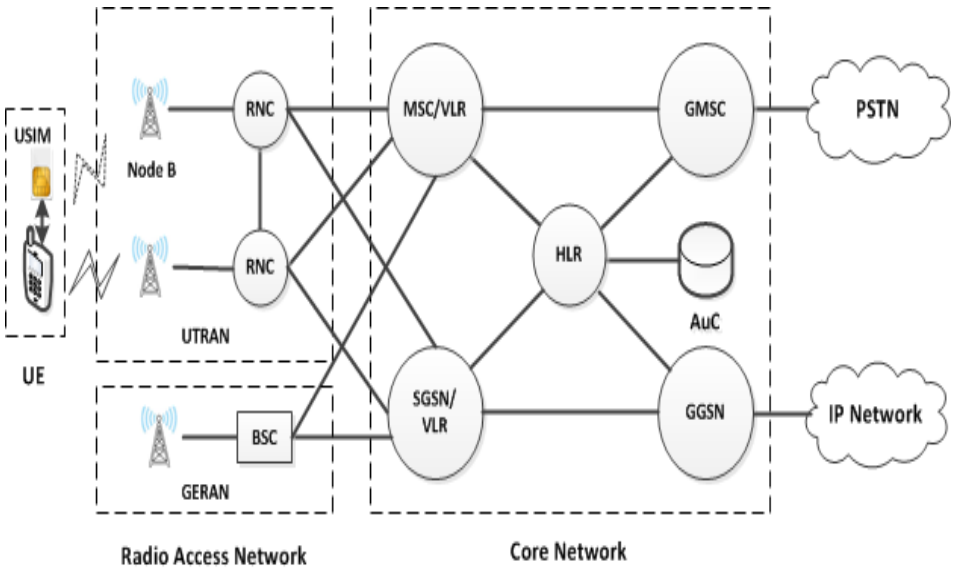

Figure 3.1: 3G systems architecture (simplified)

RAN, along with the MSC, SGSN, and their associated VLRs form part of the SN. As in GSM, the HLR and the AuC are part of the HN, and so, as in GSM, if the subscriber is roaming, the HN and SN are controlled by different MNOs.

3.2.2 4G Mobile Systems

The network infrastructure of 4G mobile systems is significantly different to that of 2G and 3G. The 4G infrastructure only supports PS data; reflecting this change, the 4G mobile system is known as the *evolved packet system* (EPS). 4G introduces a new radio access network, namely the *evolved universal terrestrial radio access network* (E-UTRAN). The radio towers in a 4G network have greater computing power, and are known as *evolved Node B* (eNB). The eNB is the only network element in the E-UTRAN, and two eNBs are connected via a direct interface facilitating fast handover between eNBs. Figure 3.2 gives a simplified view of the 4G systems architecture.

The network entities in the core network of 4G mobile systems are the *mobility management entity* (MME) and the *serving gateway* (S-GW), where the MME manages the control/signalling data and the S-GW manages the user data. This separation is a new feature in 4G. Earlier generation mobile systems supporting PS data are allowed to inter-operate with 4G systems; hence, SGSN is part of the 4G core network. The MME is responsible for authenticating subscribers, tracking the location of UEs, paging and other signalling data management. The S-GW, with the support of the *PDN gateway*, provides a data service to the subscriber. The HLR and AuC of a

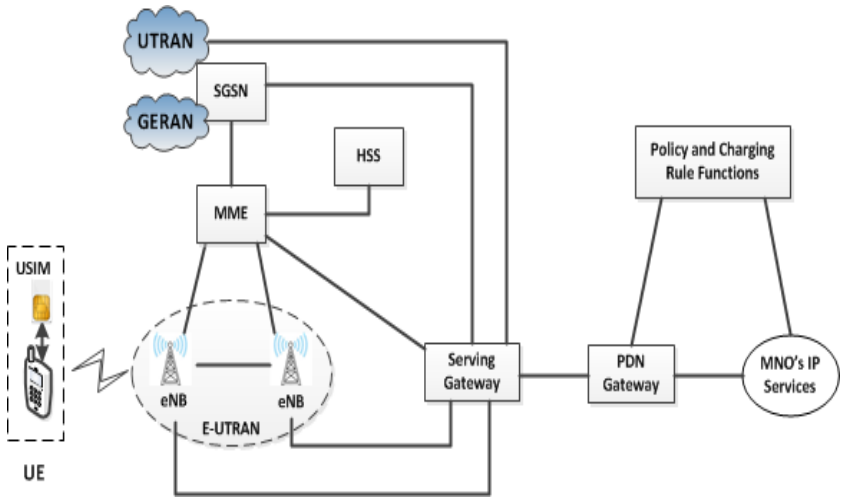

Figure 3.2: 4G systems architecture (simplified) [25, 37]

3G network are replaced with a single entity, known as the *home subscriber server* (HSS). The HSS stores mobility and service data for subscribers, and contains the functionality of an AuC. The HSS is responsible for generating authentication and key management data. The MME is connected to the HSS to support its operation, e.g. to obtain authentication data and to update location information.

The organisation of a UE remains the same as for 3G. There are no significant functional changes in the USIM. However, the 4G ME needs to support the new radio functionality to operate in the E-UTRAN. In 4G mobile systems, the E-UTRAN, along with the MME, the SGSN and the S-GW, forms part of the SN, whereas the HSS is part of the HN.

3.3 System Identities

The identities described in Section 2.3 are also used in 3G and 4G network operations. The use of an IMEI in 4G is regulated by the 4G standards which prohibit a UE from transmitting an IMEI until after a security context has been activated [93]; as a result, if equipment behaves in accordance with the standards, the IMEI should not pose a privacy threat. The 3G and 4G systems also introduce new system identities. We next briefly describe those new identities of particular relevance to user privacy.

3.3.1 P-TMSI

A P-TMSI is a temporary subscriber identity in the 3G PS domain. Like the TMSI, it is used in place of the IMSI across the air interface [23]. P-TMSIs are allocated

and managed by the SGSN. The SGSN stores the P-TMSI and its associated IMSI, and retrieves the IMSI from a P-TMSI for use in the core network. The structure of a P-TMSI is the same as that of a TMSI used in the CS domain (see Section 2.3.4). A P-TMSI is updated by the SGSN when the subscriber changes its *routing area identity* (RAI) (see Section 3.3.3 below).

3.3.2 GUTI

A *globally unique temporary UE identity* (GUTI) is a temporary subscriber identity used in the 4G EPS to provide an unambiguous identification of a UE without revealing its permanent identity. It is allocated and managed by the MME.

A GUTI is composed of a *globally unique MME identifier* (GUMMEI) and the UE's current M-TMSI for this MME. A GUMMEI contains the PLMN-ID of the network and the *MME identifier* (MMEI). An MMEI is made up of the *MME group identity* and the *MME code*, where the MME code is unique inside an MME coverage area [23]. Although the GUTI contains additional information, its function is the same as the TMSI and P-TMSI.

3.3.3 RAI

A *routing area identity* (RAI) identifies the *routing area* (RA) serviced by a single RNC. It is used in the PS domain in 3G networks. An RA is made up of a group of radio towers, analogously to an LA. However, the coverage area of an RA is a sub-division of the area covered by an LA. An RAI is composed of an LAI and a *routing area code* (RAC), where the RAC is a 8-bit field unique within an LA.

3.3.4 TAI

A 4G *tracking area identity* (TAI) identifies the *tracking area* (TA). TAs are small and non-overlapping units of area, which make up the coverage area of an MME. A TA is analogous to the location area and routing area used in GSM and 3G, respectively. A TAI contains the PLMN-ID of the network and a *tracking area code* (TAC), where the TAC is a unique code assigned by the MME.

3.4 Authentication

Authentication in 3G and 4G is performed using the AKA protocol, also known as 3GPP AKA. As in GSM, 3GPP AKA is at the core of air interface security for 3G and 4G systems. In this section we describe the 3GPP AKA protocol in detail. We

further briefly review the error reporting features of 3GPP AKA and the properties of the cryptographic functions used in 3GPP AKA.

The authentication protocols used in 3G and 4G are similar; hence, unless otherwise stated, we use 3G terminology. A detailed description of the 3G AKA protocol can be found in technical specification 3GPP TS 33.102 [31], and a description of 4G AKA can be found in technical specification 3GPP TS 33.401 [34].

3.4.1 The AKA Protocol

The AKA protocol is regularly performed between the SN and the UE for a range of reasons, as discussed in Section 2.4. It is initiated once the subscriber has been identified to the SN. In the subscriber identification phase, the UE sends its identity, i.e. its IMSI or one of the temporary identities (TMSI, P-TMSI, or GUTI depending on the access network), to the SN. On receiving the user identity, the SN first tries to determine the subscriber's IMSI; that is, if the IMSI is not sent, the SN tries to determine it from the received temporary identity using its own process. If the SN fails to determine the IMSI from the supplied temporary identity, it requests the UE to send its IMSI in cleartext, just as in GSM.

Once the serving network has learnt the IMSI, it selects the next unused element from an IMSI-specific list of AVs, stored in the serving network's VLR. If the list is empty, the SN determines the subscriber's HN by parsing the IMSI, and sends an *authentication data request* to this network. However, in 4G it is suggested not to store AVs in the SN [34, 93]; hence, in this case the SN always needs to contact the HN to request a new AV.

On receipt of such a request, the HN's AuC retrieves the secret key K for the specified IMSI, and uses it to generate a set of AVs. These AVs are generated using a set of cryptographic functions, known as $f1$-$f5$, where $f1$ and $f2$ are *message authentication code* (MAC) functions, $f3$ is a cipher key derivation function, $f4$ is an integrity key derivation function, and $f5$ is a cipher mask generating function. These functions only need to be implemented by the home network and the USIMs it issues, and none of these functions are standardised. However, the 3GPP technical specifications [35, 36] give an example set of algorithms for these functions.

Generating an AV involves the following steps (see Figure 3.3), which can be repeated as necessary.

1. A 128-bit random (or pseudorandom) value $RAND$ is generated.
2. The AuC computes the 64-bit MAC as $MAC = f1_K(RAND, SQN, AMF)$, where SQN is a 48-bit IMSI-specific sequence value and AMF is a 16-bit MNO-specific

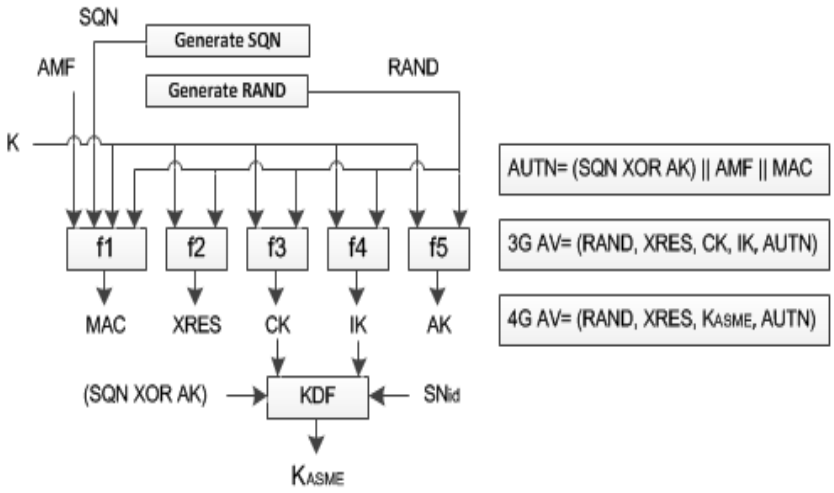

Figure 3.3: Generating an AV in the AuC

management data field.

3. The AuC computes the n-bit value $XRES$ as $XRES = f2_K(RAND)$, where the value of n is a multiple of 8 and is between 32 and 128.

4. The AuC computes the 128-bit cipher key CK as $CK = f3_K(RAND)$.

5. The AuC computes the 128-bit integrity key IK as $IK = f4_K(RAND)$.

6. The AuC computes the 48-bit anonymity key AK as $AK = f5_K(RAND)$.

7. The 128-bit authentication token $AUTN$ is generated as $AUTN = (SQN \oplus AK) \parallel AMF \parallel MAC$, where $\parallel$ denotes concatenation and $\oplus$ denotes bitwise exclusive-or.

8. A 3G AV is generated as $AV = (RAND, XRES, CK, IK, AUTN)$, and a 4G AV is generated as $AV = (RAND, XRES, K_{ASME}, AUTN)$.

   The value K_{ASME} in a 4G AV is the master session key used to derive interface-specific session keys. It is computed as $K_{ASME} = KDF_{CK,IK}(SQN \oplus AK, SN_{Id})$, where KDF is a generic key derivation function described in the 3GPP technical specifications [33, 34], and SN_{Id} is the serving network identity.

The generated AVs are then sent back to the serving network, which can then use one of them to conduct AKA. The messages exchanged among the involved parties, i.e. the UE, the serving network, and the home network, are shown in Figure 3.4, and the computations involved are shown in Figures 3.3 and 3.5.

In order to participate in AKA, the UE, in fact the USIM installed inside the UE, must possess two values:

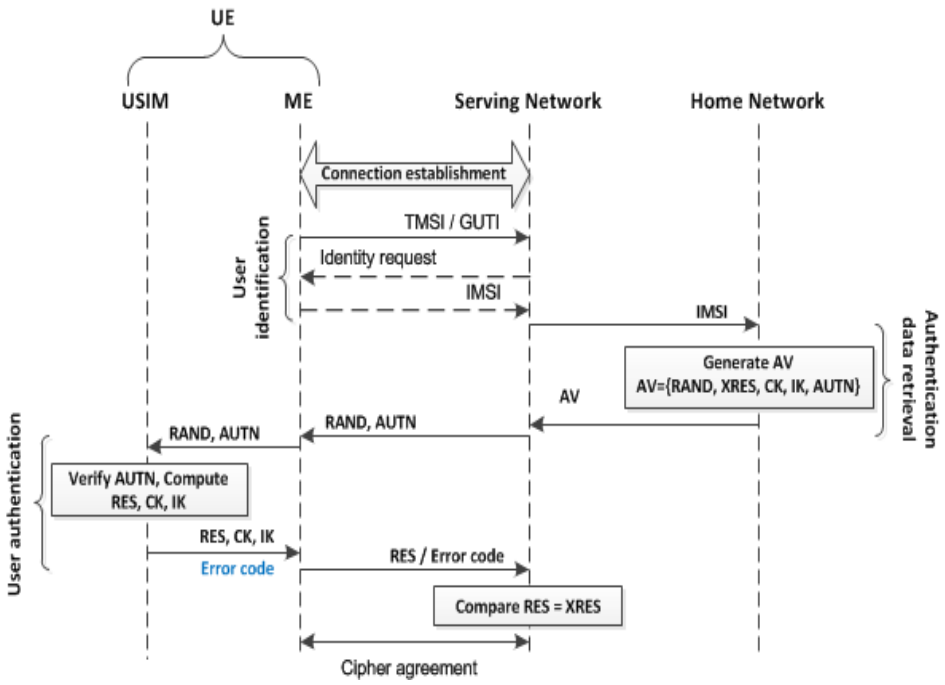

Figure 3.4: Authentication message flow in 3G mobile systems

- a long term secret key K, known only to the USIM and the USIM's HN, and
- a sequence number SQN, maintained by both the USIM and the HN.

The key K never leaves the USIM, and the values of K and SQN are protected by the USIM's physical security features. The 48-bit SQN enables the UE to verify the *freshness* of the *user authentication request*. The AK functions as a means of encrypting SQN; this is necessary since, if sent in cleartext, the SQN value would potentially compromise user privacy, given that the value of SQN is USIM-specific.

Like GSM AKA, the core of 3GPP AKA is a *challenge–response* protocol, which starts with the SN sending a *user authentication request* to the UE. More specifically, the request message contains two values, $RAND$ and $AUTN$, from the AV. The ME passes the received $RAND$ and $AUTN$ to the USIM using the $AUTHENTICATE$ command. On receipt of these two values, the USIM uses the received $RAND$, along with its stored value of K, to regenerate the value of AK using the $f5$ function, which it can then use to recover SQN. It next uses its stored key K, together with the received values of $RAND$, SQN, and AMF, in function $f1$ to regenerate $XMAC$; if the newly computed value agrees with the MAC value received in $AUTN$ then the first stage of authentication has succeeded. The USIM next checks that SQN is a 'new' value; if so it updates its stored SQN value and the network has been authenticated. If authen-

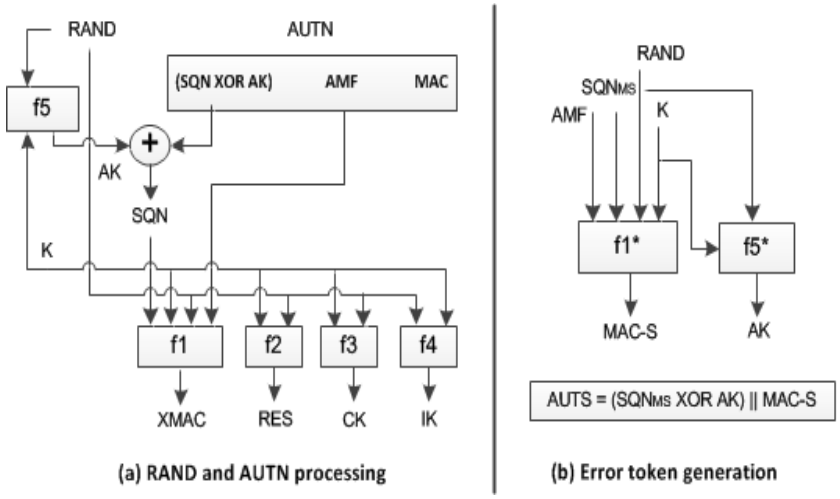

Figure 3.5: Computations in the USIM

tication succeeds, the USIM computes two cryptographic keys, i.e. *CK* and *IK*, and another message authentication code, the *RES*, and sends them to the ME. The ME stores the cryptographic keys for subsequent use, and forwards the *RES* to the network as part of the *user authentication response*. If this *RES* agrees with the value expected by the network (i.e. the *XRES* in the AV) then the UE is deemed authenticated by the SN. Figure 3.5(a) shows the computations in the USIM.

After successful authentication, the UE and the SN establish an agreed security context. The SN allocates a temporary identity, i.e. a TMSI or P-TMSI in a 3G system or a GUTI in a 4G system, and sends it to the UE via an encrypted channel.

Note that the MNO-specific value of *AMF* has no impact on the operation of 3G AKA. However, it is standardised in 4G AKA, where bit 0, the most significant bit, of *AMF* represents the *separation bit* indicating 4G authentication data, and bits 1 to 7 are reserved for future use. The remaining 8 bits are kept for MNO-specific purposes [31].

3.4.2 Error Reporting Features

The 3GPP AKA protocol could fail for a variety of reasons [26]. In this thesis we are interested in the following types of failure, namely those that involve the user authentication challenge.

- *MAC-failure*: A *MAC-failure* arises when the *MAC* value sent by the network does not match the value the USIM computes. It is reported to the SN via a signalling message. On receipt of a *MAC-failure*, the SN re-initiates AKA with a

new AV.

- *Sync-failure*: A *sync-failure* arises when the SQN value sent by the network is not greater than the USIM's stored value. It is communicated to the HN via a special token, known as $AUTS$. An $AUTS$ is constructed by concatenating a masked version of the USIM's SQN (SQN_{MS}) and a MAC computed by the USIM, known as *MAC-S* (see Figure 3.5(b)). The USIM computes *MAC-S* using the function $f1^*$ with inputs: the received $RAND$, its stored key K, SQN_{MS}, and a dummy value of AMF [31]. The function $f1^*$ is a variant of the $f1$ function used in the AKA protocol. To conceal SQN_{MS}, the USIM masks it with an anonymity key AK, computed using a variant of $f5$ known as $f5^*$. When an SN receives the $AUTS$ token, the SN forwards it to the HN with other parameters, including the value $RAND$ used in computing $AUTS$ (see Section 6.3.5 of 3GPP TS 33.102 [31]).

  When an HN receives an $AUTS$ token, it first computes AK and retrieves the USIM's SQN value from the $AUTS$ token. The HN then uses the retrieved value of SQN to verify the token. The HN adjusts its stored value of SQN to restore synchronisation with the USIM, computes an AV with the new value, and sends this AV to the SN for use in AKA.

3.4.3 Properties of the Cryptographic Functions

The AKA protocol relies on the cryptographic functions $f1$–$f5$, $f1^*$, and $f5^*$, implemented in both the USIM and the AuC of the home network. We use the functions $f1$ and $f5$ in a modified way in designing our schemes; hence, we next briefly describe the properties of these functions.

The function $f1$ generates a MAC value on the inputs using the key K. The function has the property that it must be computationally infeasible to derive the key K from knowledge of the inputs and the output of $f1$ [32]. Example of such functions include CBC-MAC or HMAC [107].

The function $f5$ outputs a derived key when given as input a master key K and a 128-bit string. Like $f1$, it must be computationally infeasible to derive the master key K from knowledge of the input string and the output of $f5$ [32].

3.5 The USIM

A USIM is a smart card application residing in a UICC that is employed by a UE to access 3G and 4G mobile networks. It stores a wide range of data in EFs about

the subscriber, the services supported, and the network, including the subscriber's IMSI. We described the storage structure of a USIM application in Section 2.6.1. The mandatory information storage requirements for a USIM application are defined in 3GPP TS 21.111 [20]. The USIM links the subscriber to the issuing MNO, contains the cryptographic functions required to execute the AKA protocol, and forms the basis for the air interface security features.

The USIM application exchanges information with the ME using *command–response* pairs, much like a SIM, as discussed in Section 2.6.2. The structure of the command and response APDUs are similar to those exchanged with a GSM SIM (see Figure 2.7). However, the value of the class byte in the USIM commands is different from that of the SIM commands, and a USIM supports a 3G-specific set of commands identified by the 'INS' byte. Further details of the USIM can be found in technical specification 3GPP TS 31.102 [28].

One of the mandatory requirements for 3G mobile systems is to support the 2G mobile services transparently to the users [22]; hence, a 3G UE must be able to support the 2G radio access protocols. In this section we briefly describe the USIM features which allow a USIM to support 2G functionality. We also describe another USIM feature, known as the *USIM application toolkit*, that is of key importance in designing the schemes we describe later in the thesis.

3.5.1 Modes of Operation

Support for 2G access networks is a mandatory requirement for a USIM application. A 3G ME, i.e. an ME supporting both 2G and 3G, must select the USIM application when it is present in the UICC regardless of the radio access technology. This implies that a 3G UE will never use a SIM application if it is present on the UICC, but it nevertheless must support the 2G services transparently to the subscriber. This requires the USIM application on the UICC to support 2G functionality to enable it to provide service when its host ME roams in a 2G mobile network.

A USIM application stores the 2G-specific data in its rooted file tree, i.e. in ADF_{USIM}. More specifically, the dedicated file $DF_{GSM-Access}$ contains the necessary data (see Figure 2.6). Since the ME can only use the 3G command set to interact with the UICC, the UE never runs GSM AKA when it roams in a 2G network; instead the USIM operates in something called *virtual 2G mode* to run the GSM authentication protocol. In virtual 2G mode, the USIM receives only the value of $RAND$ as the authentication parameter. On receiving $RAND$, the USIM first executes $f2$ to generate the RES, $f3$ to compute the CK, and $f4$ to compute the IK. Subsequently, it applies the conversion function $c3$ to compute the 2G K_c from the CK and IK, the conversion function

$c2$ to generate the 2G *SRES* from the *RES*, and returns *SRES* and K_c to the ME. The USIM switches to virtual 2G mode on receiving a particular command parameter depending on the current radio access network. The conversion functions, i.e. $c2$ and $c3$, are described in annex B of technical specification 3GPP TR 31.900 [17].

The other possible USIM operation mode is known as the *3G + K_c mode*, where the 2G ciphering key K_c is included in the USIM's authentication response, and the ME picks the relevant values depending on the current radio access network. The key K_c is computed using the conversion function $c3$ as described above.

The fact that a USIM supports the above described modes of operation in addition to its usual *3G mode* is disclosed in its service table. The USIM needs to support service number 27, known as *GSM access*, to operate in *3G + K_c* mode. Generally this mode is always active if the service *GSM access* is available in the USIM. Alternatively, USIM service number 38, known as *GSM security context*, must be activated in the USIM to operate in virtual 2G mode. The ME learns the set of services supported by the USIM during USIM initialisation.

3.5.2 USIM Application Toolkit

Analogously to the SIM Application Toolkit, the *USIM application toolkit* (USAT) is a service operating across the interface between the USIM and the ME. It provides a mechanism for a USIM application to initiate an action to be undertaken by the ME. A USIM that supports the USAT is known as a *proactive UICC*. A proactive UICC provides a set of commands, known as the *proactive commands*, which allow the USIM to operate with an ME. The proactive commands are grouped into a variety of classes, and an ME supporting an individual class must support all the commands in that specific class.

A proactive UICC extends the services provided by a *proactive SIM* described in Section 2.6.3. The execution steps of a proactive command are exactly same as that of the SAT command, and so the description of SAT in Section 2.6.3 also applies to USAT. Further details of USAT commands can be found in the relevant 3GPP standards [29, 89].

3.6 Network Activities

We next re-examine the network activities described in Section 2.5 in the context of 3G and 4G mobile systems.

3.6.1 IMSI Attach

The IMSI attach procedure in 3G and 4G is similar to the attach procedure in GSM (see Section 2.5.1). In both systems, when a UE is switched on, it listens for a specific *SYSTEM INFORMATION* message containing the PLMN-ID of the radio tower. On receiving the message, the UE compares the PLMN-ID values in the message with the list of permitted networks stored in its USIM. If the UE finds a match, it initiates the IMSI attach process.

In 3G systems the UE requests the SN to initiate a *location update* procedure (see Section 3.6.2 below), indicating an *IMSI attach* in the *update type* field of the request message. This location update registers the UE in the network, and updates the LAI and RAI of the UE simultaneously. The IMSI attach procedure in 3G is described in Section 4.4.3 of 3GPP TS 24.008 [26].

In 4G systems the UE sends an *attach request* message to the MME, where the message includes an IMSI or an old GUTI, the old GUTI type, and the attach type. The value of the attach type indicates an IMSI attach. On receiving such a message, the MME authenticates the UE, updates the location information, and sends the UE a response. The IMSI attach procedure in 4G is described in Section 5.3.2 of 3GPP TS 23.401 [25].

3.6.2 Location Update

The location update procedures in 3G and 4G are both similar to the corresponding procedure in GSM (see Section 2.5.2). In both 3G and 4G, the UE sends its stored temporary identity, last visited location identity, and the type of location update to the SN. However, the events that can initiate a location update differ between 3G and 4G. For example, in 3G, if a subscriber is attached to the PS domain, a change in the RAI of the subscriber's roaming area initiates a location update procedure, as does a change in the value of the LAI associated with the subscriber. In 4G, the location update procedure is initiated when the subscriber roams in a TA with a TAI that is not in the list of TAIs currently stored by the UE. A detailed list of possible triggers for a location update in 4G can be found in Section 5.3.3 of 3GPP TS 23.401 [25].

On receiving the location update request, the SN runs the AKA protocol, enables the security context, and sends a new temporary identity to the UE; it then initiates a location update with the subscriber's HN, involving the following steps.

1. The serving MSC (in 3G CS connections), SGSN (in 3G PS connections), or MME (in 4G), sends its identifier and the IMSI in a *location update* request to the subscriber's HLR (in 3G) or HSS (in 4G).

2. On receiving the message, the HLR or HSS updates the location information for the subscriber identified by the IMSI, and sends a *cancel location* request to the 'old' MSC, SGSN, or MME for this subscriber.

3. The 'old' MSC, SGSN, or MME sends an acknowledges for the location cancellation to the HLR or HSS.

4. The HLR or HSS sends an acknowledges for the *location update* request to the initiating MSC, SGSN, or MME.

The SN sends a *location update accept* or *location update reject* message to the UE. Further details of location management in 3G and 4G can be found in 3GPP TS 23.012 [15] and 3GPP TS 23.401 [25].

3.6.3 Paging

The paging process in 3G and 4G is similar to that in GSM (see Section 2.5.3). In 3G, paging is controlled by the 3G MSC and SGSN in the same way as it is managed by a 2G MSC in GSM. In both 3G and 4G, the paging message contains either an IMSI or a temporary identity. The 3G system defines two types of paging messages, known as *paging type 1* and *paging type 2* (see Sections 10.2.20 and 10.2.21 of 3GPP TS 25.331 [27]), where *paging type 2* messages are integrity protected when transmitted across the air interface. However, paging type 1 messages are not integrity protected (see Section 6.5.1 of 3GPP TS 33.102 [31]). Further details of paging in 3G can be found in the 3GPP standard [27].

Paging also serves the same purposes in 4G systems, and is used by the MME in similar ways as in 3G. However, 4G introduces *smart paging*, which enables the MME to broadcast a paging message in an area covered by a single eNB. Paging in 4G mobile systems is described in Section 5.3.2 of 3GPP TS 36.331 [38].

3.6.4 Mobile Terminated Services

As in GSM (see Section 2.5.4), MT services in 3G and 4G are supported by the paging procedure. In both systems, the core network first determines the responsible MSC (in 3G CS connections), SGSN (in 3G PS connections), or MME (in 4G), for the target subscriber with the help of the subscriber's HLR (in 3G) or HSS (in 4G) to deliver the MT service. Next, the responsible entity uses the paging process to establish a connection with the target subscriber, which is followed by subscriber authentication, security context and service setup, and delivery of the pending service to the subscriber.

3.7 Synchronisation of Temporary Identity

In GSM, 3G and 4G, a pseudonym (temporary identity) is normally used in place of the permanent subscriber identity (the IMSI) in communications across the air interface, whereas communications in the core network are based on the IMSI. The relationship between the subscriber's temporary identity and its IMSI is maintained by the VLR in GSM and 3G systems, and by the MME in 4G systems. Synchronisation of the temporary identity between the UE and the SN is necessary for successful operation. In this section we briefly review how subscriber temporary identities are managed, and how identity desynchronisation is addressed by the network.

Temporary identities are allocated by the SN, and are sent to the UE. On receiving the identity, the UE updates its stored temporary identity, and acknowledges the event. The SN waits for a predetermined time period for an acknowledgement from the UE. If the SN does not receive an acknowledgement, it records both the old and the new temporary identities against the corresponding IMSI. While this scenario holds, the SN uses the subscriber's IMSI when communicating with the UE across the air interface regarding the provision of MT services. After successful connection establishment, the SN renews the temporary identity, that is, it deletes the association between the IMSI and the existing temporary identity to allow the released identity to be allocated to another UE, and allocates a fresh temporary identity.

For mobile-originated communications, the SN allows the UE to identify itself using either the old or the new temporary identity. This helps the SN determine the temporary identity that is currently stored in the UE. The SN subsequently deletes the association between the other temporary identity and the IMSI to allow this temporary identity to be allocated to another user [31].

Although details of how temporary identities are managed are up to the MNO, it cannot guarantee synchronisation of the temporary identity between the UE and the SN. For example, if a UE becomes unresponsive to network queries for a significant period of time, the SN may reallocate the previously allocated temporary identity to another subscriber; hence, air interface protocols allow the correct IMSI to be recovered from a desynchronised temporary identity.

The SN assumes loss of identity synchronisation when it receives repeated *MAC-failure* messages from AKA attempts. To recover from identity desynchronisation, the SN requests the UE for its IMSI in cleartext. On receiving the IMSI, the SN first checks whether there has been a desynchronisation of the temporary identity; if so, the SN runs AKA and allocates a new temporary identity.

Part II

Security and Privacy Issues in 2G

Overview

Part II describes the security and privacy threats arising from use of GSM. It further presents novel schemes to support mutual authentication, and to improve air interface user privacy in GSM networks. It contains the following three chapters.

- *Chapter 4* gives an overview of known security and privacy issues arising from the lack of mutual authentication and disclosure of the permanent subscriber identity in GSM. It also briefly discusses previous attempts to address these issues.
- *Chapter 5* describes and analyses a novel scheme to provide mutual authentication in GSM without affecting the serving networks or phones.
- *Chapter 6* describes and analyses a scheme to improve air interface user privacy in GSM, that does not require modifications to the serving networks or phones.

Chapter 4

Security and Privacy Issues in GSM

4.1 Introduction

The GSM mobile system was designed back in the 1980s, and hence the threat analysis underlying its design reflected the types of threat that were deemed realistic at that time. Several decades later, it is hardly surprising that threats deemed unrealistic back then are now practical realities, meaning that GSM possesses a range of significant security and privacy vulnerabilities. In this chapter we briefly describe the known GSM security and privacy vulnerabilities arising from its lack of network authentication and the disclosure of the permanent subscriber identity. We further describe a range of possible modifications that have been proposed to try to address these threats.

The chapter is organised as follows. Section 4.2 briefly reviews the security and privacy features of the GSM air interface. In Section 4.3 we give a classification of the possible types of attack on a mobile system. Sections 4.4 and 4.5 describe the relevant security and privacy threats in GSM. In Section 4.6 we review previous attempts to address the security and privacy threats described in Sections 4.4 and 4.5. Finally, Section 4.7 concludes the chapter with a discussion of the motivation for the work presented in Chapters 5 and 6.

4.2 Security and Privacy Features

The GSM system introduced security and privacy support, something not present in the first generation of mobile systems. It provides the following security and privacy features for the GSM air interface:

- IMSI confidentiality,
- IMSI authentication,
- user data confidentiality on physical connections,
- connectionless user data (short message) confidentiality, and
- signalling information element confidentiality.

We next briefly describe these features. Further details can be found in GSM 02.09 [11].

4.2.1 Subscriber Identity (IMSI) Confidentiality

The purpose of IMSI confidentiality is to avoid an interceptor of mobile traffic being able to identify which subscriber is using a given resource on the radio path; hence, it protects subscribers against possible tracking. The provision of this feature implies that the IMSI should not be transmitted in cleartext in any signalling message on the radio path.

As discussed in Section 1.2, instead of using the IMSI, a TMSI is used to identify a mobile subscriber on the radio path to provide subscriber pseudonymity. However, there are certain special circumstances in which the IMSI is transmitted across the air interface. Note that an IMSI is always sent across an unencrypted channel since the key necessary for channel encryption is not available to the network until the network knows which MS it is communicating with. This violates user privacy and limits the effectiveness of the IMSI confidentiality feature. This potential breach of subscriber privacy occurs in the following circumstances.

- *When an ME is switched on*: In this case there is neither a valid temporary identity nor security context information available to the MS; hence, the MS has no option but to use its IMSI to identify itself to the SN.
- *When an SN fails to retrieve the IMSI*: Since the SN must know the subscriber's IMSI to authenticate, as discussed in Section 2.4, the SN maintains a mapping between temporary identities and corresponding IMSIs. However, an SN might fail to retrieve the IMSI for a received temporary identity for a range of reasons, e.g. a crash or malfunction of the VLR database, or a faulty implementation of temporary identity management. In such scenarios the SN requests the MS for its IMSI, and the MS responds with the IMSI.

- *When an 'old' SN fails to provide the requested IMSI*: When an MS initiates a location update procedure because of a change in LAI, the SN forwards the received temporary identity to the subscriber's 'old' MSC requesting the subscriber's IMSI (see Section 2.5.2). However, the 'old' MSC could fail to provide the requested IMSI, e.g. if the 'old' MSC has already deleted its record of the subscriber, or if the communication between the SN and the 'old' SN fails because of a communication error. In such cases the SN requests the MS for its IMSI, and the MS responds with the IMSI.

- *When the temporary identity shared by the MS and the SN is desynchronised*: In this scenario the SN retrieves an IMSI which is not that of the communicating MS. This causes a failure in subsequent operations (see Section 3.7). To recover from identity desynchronisation, the SN requests the MS to send its IMSI across the air interface.

- *When an SN broadcasts a paging message*: The scenarios described above cover legitimate network operations that require an MS to disclose its IMSI across the air interface. In addition, an SN sends an IMSI across the air interface in paging messages, more specifically in *paging type 1* messages (see Section 2.5.3).

4.2.2 Subscriber Identity (IMSI) Authentication

Subscriber identity (IMSI) authentication ensures that the subscriber identity (IMSI or TMSI) transferred by the MS within the identification procedure across the radio path, is as claimed.

IMSI authentication is triggered by the MSC of the SN. Several scenarios exist in which an MSC will perform an authentication, depending on whether the TMSI or IMSI is used for identification, and whether the TMSI can be used to retrieve an AV as necessary to perform the authentication procedure. Details of the authentication process are given in Section 2.4.

4.2.3 Data Confidentiality

The purpose of this feature is to ensure privacy of user data when sent across traffic channels (i.e. voice communications data), user data sent via signalling channels (i.e. short messages), and user-related signalling information elements included in signalling messages.

Data confidentiality is achieved by the use of encryption. More specifically, the function $A5$, a GSM-specific stream cipher algorithm, is used to encrypt user data

before it is sent across the air interface. The encryption key, K_c, is refreshed as part of every successful authentication. Several versions of $A5$ exist, and a negotiation between the MS and the base station of the SN is carried out to decide which version of $A5$ to use.

Confidentiality of signalling information elements applies to certain fields of the signalling messages exchanged between an MS and a base station. The user-related signalling information element, i.e. the IMSI, TMSI or IMEI, used to establish the connection is not protected; however, the IMSI, IMEI, calling subscriber directory number (in mobile terminating calls), and the called subscriber directory number (in mobile originated calls) are protected when sent after connection establishment [11]. To ensure identity confidentiality, the TMSI is transferred via an encrypted channel at allocation time.

4.3 Types of Attacker/Attack Modes

Before describing the relevant attacks, in this section we briefly outline the mobile system attack model of relevance to this thesis, as described by Shaik et al. [147]. In the attack model, the following three attack modes are based on the attacker's capability. We use the term *attacker* and *adversary* interchangeably in the rest of the thesis.

- *Passive*: An attacker in this mode is able to silently eavesdrop on over-the-air (radio) transmitted data. To achieve this, the passive adversary has access to a hardware device, for example a *universal software radio peripheral* (USRP)[1] and associated software as necessary to observe and decode radio-transmitted messages.

- *Semi-Passive*: A semi-passive adversary is, in addition to passive monitoring, able to trigger the transmission of signalling messages to subscribers using interfaces and actions that are legitimately available in mobile systems. For example, a semi-passive adversary can trigger the transmission of a paging message to a subscriber by sending a message or initiating a call to the target subscriber. The adversary is assumed to be aware of an identity of a subscriber, such as a Facebook profile or a mobile phone number. A semi-passive adversary is analogous to the *honest-but-curious* or *semi-honest* adversary model used for cryptographic protocols [99].

- *Active*: An active adversary can set up and operate a fake base station or radio tower to establish unauthorised communications with a subscriber. The capabili-

[1] `http://www.ettus.com/product/details/UB210-KIT`

ties required for active attacks include knowledge of mobile network configurations and access to appropriate hardware, e.g. a USRP, that can be used to impersonate an SN to the target mobile device. An active adversary is analogous to the *malicious* adversary model used for cryptographic protocols [99].

4.4 Fake Base Station Attack

In GSM, although the MS is authenticated to the SN, the SN is not authenticated to the MS (see Section 2.4). That is, GSM AKA provides unilateral, rather than mutual, authentication. This allows the possibility of attacks in which an active adversary masquerades as a base station of an SN to one or more MSs. This is known as a *fake base station attack*.

Active attacks involving the impersonation of network elements were considered when GSM was originally designed, but were not deemed to be worth addressing. The perceived complexity of building base station devices was assumed to be so high that the risk arising from the threat was assessed as being rather small. Further, the fact that traffic exchanged between the base station and the MS is encrypted reduces the risks arising from the lack of mutual authentication (see also [138]).

However, a level of threat does remain, not least because of the following facts.

- The cost of base station devices has fallen rapidly, and 'testing' devices capable of emulating a genuine base station are readily available [166, 168].
- The use of encryption on the air interface is completely controlled by the base station, and some networks do not 'switch on' data encryption.

These factors make the use of a false base station a serious threat to the claimed security and privacy properties of GSM. That is, GSM systems possess a wide range of security and privacy vulnerabilities. We next explore the precise implications of the fake base station threat by describing selected examples of possible active attacks on the radio path.

4.5 Threats Arising from Base Station Impersonation

4.5.1 Man-in-the-Middle Attack

In a *man-in-the-middle* attack the attacker secretly relays and possibly alters the communication between two parties who believe they are directly communicating with each

other. Possible man-in-the-middle attacks on the GSM air interface have been extensively discussed [53, 56, 125, 126, 130, 136, 138, 160]. Before outlining one possible man-in-the-middle attack previously discussed by Mitchell [130] and Pagliusi [136], we briefly explain the resources assumed to be available to the attacker. Note that, apart from the attack described immediately below, the attacks discussed in Sections 4.5.2–4.5.5 are also all examples of this general class of attack.

The attacker must have a device capable of emulating a base station. In addition the attacker should have a valid subscription to a GSM network and an MS device, presumably incorporating the attacker's SIM, able to communicate with a genuine base station. The attacker's MS should be integrated with the emulated base station.

The fact that the SN always decides whether or not to enable encryption makes it possible for a fake base station to act as an intermediary between an MS and a genuine network. The attack steps are as follows.

1. The attacker uses its fake base station to capture the target MS. That is, the target MS, presumably belonging to an individual whose calls the attacker wishes to intercept, registers with the attacker's fake base station, believing it to be a base station belonging to a legitimate network.

2. When authentication needs to take place between the captured MS and the fake base station, the fake base station sends an arbitrary *RAND* value to the MS, and can ignore the SRES returned in response. The fake base station does not enable encryption on the link to the MS, so the fact that it does not know the encryption key does not matter.

3. Since the MS will not be encrypting air interface traffic, the fake base station can detect when the MS makes a call, and can also read the dialled digits. The fake base station then uses its integrated MS to talk to the genuine network to place a call to the same destination using the IMSI or TMSI belonging to the attacker.

4. All traffic sent via the fake base station (the man-in-the-middle) is simply relayed between the target MS and the attacker's MS. If the genuine network chooses to enable encryption, then the fake base station can communicate with it successfully since it is using its own SIM for this leg of the communications (see Figure 4.1).

5. The fake base station can now seamlessly listen to all the voice traffic sent to and from the victim MS, and can monitor all the associated signalling information elements, including the called and calling numbers.

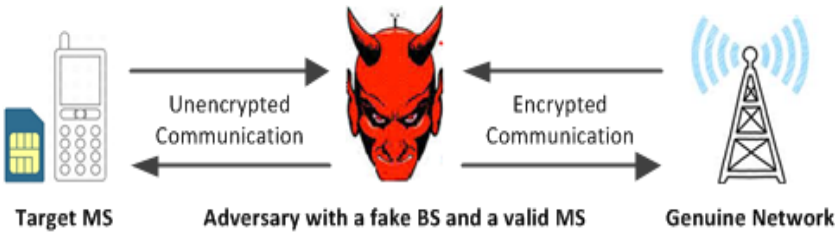

Figure 4.1: Possible scenario for the man-in-the-middle attack

If the genuine SN enables data encryption, the attacker has to pay for all the calls which it intercepts; however, this may be a small price to pay for the confidential conversations that it intercepts.

Mitchell [130] further described similar attacks with variations in the required resources, e.g. an attacker without a valid mobile subscription could spoof an answer to a called number, redirect a call to any destination chosen by the attacker, or set up prank calls to a target subscriber.

4.5.2 Barkan-Biham-Keller Attack

The fact that the network always decides whether or not to enable encryption and also chooses the $A5$ variant to be used allows the well known *Barkan-Biham-Keller* attack, [52, 53]. This attack enables recovery of the encryption key; this then enables unlimited interception of phone calls. The attack takes advantage of the following key facts.

- The encryption algorithm $A5/2$ is very weak, and can be broken in real-time [52, 96].
- The SN decides which encryption algorithm to use.
- The same key K_c is used with all three variants ($A5/1$, $A5/2$, $A5/3$) of the GSM encryption algorithm.
- GSM AKA allows reuse of an authentication challenge $RAND$.

One possible scenario for the attack is as follows. Suppose an attacker intercepts the GSM AKA exchanges between the SN and an MS, notably including the $RAND$, and also some of the subsequent encrypted voice exchanges involving that MS. Suppose also that the MS is subsequently switched on within the range of a fake network operated by the attacker. The fake network inaugurates the AKA protocol with the MS and sends the previously intercepted $RAND$, causing the SIM in the MS to generate the same K_c

as was used to encrypt the previously intercepted data. The MS responds with *SRES*, which the fake network ignores, and the fake network now enables encryption using $A5/2$. The MS will now send data to the SN encrypted using $A5/2$ and K_c; because of certain details of the GSM protocol, the plaintext data will contain predictable redundancy. The fake network now takes advantage of the weakness of $A5/2$ to recover K_c from the combination of the ciphertext and known redundancy in the corresponding plaintext. The key K_c can now be used to decrypt all the previously intercepted data, which may have been encrypted using a strong encryption algorithm such as $A5/3$.

4.5.3 IMSI Catching Attack

IMSI catching attacks have been discussed previously in a number of research papers — see, for example, [56, 68, 138, 151]. The fact that there is a provision to allow a base station to request an MS to send its IMSI in cleartext across the air interface means that there is a very straightforward way for a fake base station to compromise IMSI confidentiality. The fake base station simply sends the *user identity request* message to the MS. In response, the MS sends the IMSI in a *user identity response* message (as always, the IMSI is sent via an unencrypted channel). This is known as an *IMSI catching* attack. A fake base station associated with such attack is widely known as an *IMSI catcher*. The first case of the use of an IMSI catcher recorded in the literature [151] occurred in 1996 in Germany.

Once an IMSI has been obtained by an unauthorised party it can be used to determine the associated phone number, e.g. with the help of the MNO to which the IMSI belongs. A very similar IMSI catcher attack also works against 3G and 4G networks, as we discuss in Chapter 7.

4.5.4 User Linkability Attack

This attack exploits the fact that GSM AKA allows reuse of an authentication challenge. Mitchell [130] discusses this threat, which stems from the observation that if a base station sends a particular *RAND* value to an MS, then the MS always return the same *SRES* value in response.

Suppose an attacker with a fake base station has at some time in the past intercepted a genuine (*RAND*, *SRES*) pair for a particular MS; then the attacker can determine whether an MS is the same as the previously identified MS by sending it the same *RAND* value and observing the response. That is, the authentication response can be used to distinguish between MSs. This allows an attacker to trace the target subscriber within an attacker's coverage area as long as the target subscriber uses the same SIM.

4.5.5 IMSI Paging Attack

IMSI paging attacks have been discussed by a number of authors [48, 49, 98, 117, 147]. This attack exploits a specific type of signalling message, known as a *paging* message (see Section 2.5.3); more specifically, it exploits a *paging type 1* message. Such messages are sent from the SN to all mobile devices in a particular area, and contain either an IMSI or a TMSI. If an MS detects such a message containing its IMSI or its current TMSI then it responds with a message containing its current TMSI.

The key fact which makes the IMSI paging attack feasible is that the paging messages are not cryptographically protected; hence an active adversary can introduce spurious paging messages into the network. This can be used both to detect the presence of an MS with a specific IMSI in a location, and to learn the current TMSI for this device.

Since there exist scenarios in which an SN fails to identify the correct TMSI of a subscriber, e.g. when an SN does not receive an acknowledgement to a TMSI allocation message, the SN is obliged to use a subscriber's IMSI to support MT services (see Section 3.7); thus, the use of type 1 paging message cannot be avoided. However, the SN can limit the use of type 1 paging messages to minimise the threat to subscriber privacy arising from *passive adversaries*. A number of studies [98, 117] have shown that in real-world networks the vast majority of paging requests are of type 1. Using this fact, Kune et al. [117] demonstrate how GSM paging messages enable a passive adversary to learn the subscriber location.

4.6 Fixing GSM

The lack of network authentication in GSM enables a range of fake base station attacks, and so adding network authentication could mitigate many of the vulnerabilities described above. In this section we review key examples of (a) prior art that describes ways of adding network authentication to GSM, and (b) previously proposed techniques for protecting the privacy of the IMSI.

4.6.1 Inclusion of Network Authentication

The possibility of adding network authentication to GSM has been considered by the 3GPP technical specification group. Two 3GPP TSG documents [81, 82] proposed the introduction of network authentication into GSM. Ericsson [82] proposed transferring authentication responsibility to the terminal by implementing the core of the 3G AKA protocol entirely in software. However, the scheme in turn raised other security threats.

Neither of the proposals were adopted, presumably because of cost/feasibility issues.

Apart from the work within 3GPP, many authors have described possible ways of improving the GSM authentication protocol [46, 67, 70, 90, 91, 116, 119, 122]. However, most of them involve completely redesigning GSM AKA, requiring changes to all the SNs as well as all the deployed phones. We next review key examples of this work.

Kumar, Shailaja and Kavitha [116] propose an identity-based mutual authentication scheme for GSM. Although their scheme protects against replay and man-in-the-middle attacks, it introduces identity-based cryptography [148] into a system which currently only uses symmetric cryptography; unsurprisingly, the scheme requires modifications to the operation of all entities involved in authentication, i.e. the SIM, the ME, the SN, and the HN. Agarwal, Shrimali and Lal Das [46] describe a somewhat similar scheme that provides mutual authentication using identity-based cryptography.

Lo and Chen [122] present a mutual authentication scheme for GSM using public-key cryptography [141]. However, this again requires significant infrastructural changes, including modifications to the operation of the SIM, ME, SN, and HN.

Fanian, Berenjkoub and Aaron Gulliver [90, 91] propose a mutual authentication scheme for GSM following the *timed efficient stream loss-tolerant authentication* (TESLA) multicast authentication protocol [137]. The scheme involves completely redesigning the authentication protocol, and hence requires changes to the operation of all the GSM entities.

Choi and Kim [70] propose an authentication scheme for GSM supporting mutual authentication. Although the scheme makes use of the existing cryptographic techniques, it involves completely redesigning the message flows; that is, the MS generates a *RAND*, computes an *SRES*, and sends them to the network. Like the earlier schemes, this scheme also modifies the operation of the SIM, phone, SN and HN.

Lee, Hwang and Yang [119] propose an extension of GSM authentication to include network authentication. The scheme relies on symmetric cryptography and the $A3$ function, available to both the SIM and the AuC of the HN, to perform network authentication. It makes the use of an MS-generated time-stamp value T which is sent to the HN via the SN. On receiving T, the AuC of the HN verifies that it is current; if so, the AuC computes a network authentication token as $A3_K(T)$, and sends the computed token along with the required authentication data to the SN. The SN sends the token and the earlier received T with the usual authentication challenge to the MS. The MS verifies the received value of T, computes the network authentication token in the same way as it was computed in the HN, and compares the computed token with the received token. If they match, the network is deemed authenticated by the MS. The HN also computes a temporary master key and sends it to the SN. The SN

uses this master key to compute subsequent AVs, and authenticates the MS using the existing authentication process; that is, mutual authentication is performed when the MS first joins the SN. All subsequent authentications with the same SN are performed without ensuring mutual authentication, which is clearly a drawback of the proposed scheme.

Chang, Lee and Chang [67] proposed a modified version of the Lee-Hwang-Yang scheme addressing the drawback mentioned above. However, like the Lee-Hwang-Yang scheme, the Chang-Lee-Chang proposal involves modifying the protocol messages to include additional parameters, i.e. a time-stamp and a token; hence, both schemes require modifications to the operation of the ME and SN.

Since all these schemes require significant changes to the operation of all the entities in a GSM network, they are most unlikely to be deployed in practice. Of course, these schemes might be worth considering for use in future networks, but how authentication might work in 5G networks and beyond is outside the scope of the work described here.

4.6.2 IMSI Privacy Protection

The problem of the lack of robust user identity privacy in mobile networks is more than two decades old, and has been discussed extensively. A wide range of research trying to address the problem exists; although recent work mostly focuses on the privacy threats to 3G and 4G, there is a significant body of established research addressing GSM privacy; we summarise below key elements of that work.

In the mid 1990s, Samfat, Molva and Asokan [143] addressed the conflicting requirements of untraceability and disclosure of identity during authentication in mobile networks. They proposed the use of an alias which is only understandable to the user's HN in order to hide the user's real identity from both eavesdroppers and the SN. Their scheme uses probabilistic public-key encryption of the real identity for alias computation. Since GSM does not support public-key cryptography, adoption of the scheme would introduce significant changes to the GSM architecture. A similar scheme was described by Samfat and Molva [142] with the goal of protecting the confidentiality of the subscriber IMSI.

Herzberg, Krawczyk and Tsudik [102] and Ateniese et al. [50] discuss issues with anonymity and location privacy in mobile networks. They propose a range of anonymity approaches intended to conceal the real identity of an user, including what they call a time-based alias computation, a home-centric approach, and a public-key based approach. Of particular relevance here is the home-centric approach, in which the HN computes aliases for its users and sends them to the user in user authentication messages. Aliases are computed by encrypting a combination of a (variable) salt and the

'real' identity, which means that aliases are longer than the real identities.

Lee, Hwang and Yang [118] propose a privacy enhanced scheme for GSM to improve user identity and location privacy. Their scheme avoids the need for the IMSI to ever be sent across the air interface. It introduces an HN-managed TMSI (equivalent to an alias), which contains the subscriber's PLMN-ID and an encrypted version of the subscriber's IMSI, concatenated with a time-stamp value. Although the use of symmetric encryption is suggested, it is not clear how the HN identifies the correct key to decrypt the IMSI embedded in the TMSI. A new TMSI (HN-managed) is allocated in every location update.

The problem of loss of user privacy arising from cleartext IMSI transmission across the air interface also arises in 3G and 4G networks, and this is discussed further in Chapter 7.

Most significantly from the perspective of this thesis is the fact that all the previous proposals to enhance IMSI privacy protection require major changes to the operation of all GSM networks, including modifications to the SN, HN, ME and SIM. Making such fundamental changes would be hugely costly and very difficult to manage, and this makes it most unlikely that any of the schemes will ever be deployed in practice.

4.7 Research Motivation

GSM has been criticised for its weak security design, and a range of security flaws have been uncovered after its deployment. As discussed in Sections 4.4 and 4.5, GSM is vulnerable to a wide range of threats because of the absence of network authentication. Nevertheless, GSM continues to be widely used.

Despite the introduction and deployment of 3G and 4G mobile systems, which rectify the authentication shortcoming by providing mutual authentication between phone and network, GSM remains of huge practical importance worldwide and is not likely to be phased out for many decades to come. As a result, finding a way to incorporate network authentication into GSM in a way which can be deployed in practice is of great practical importance. Section 4.6.1 described some of the key existing proposals for adding network authentication to GSM AKA; unfortunately, they all require significant modifications to the air interface protocol, which would require changes to the operation of all the serving networks as well as all the deployed phones. As noted in Section 4.6.1, it seems likely that making the necessary major modifications to the operation of the air interface after deployment is infeasible in practice. This observation motivates the investigation of possible ways of including network-to-phone authentication in GSM in a way that is completely transparent to the intermediate network infrastructure, and

hence only requires SIMs and the HN to be upgraded. Since both the SIM and the HN are managed by a single MNO, such a solution can be rolled out piecemeal with no impact on the existing global infrastructure. We describe and analyse such a scheme in the next chapter.

Similarly, all the previously proposed solutions to the IMSI catcher problem, as described in Section 4.6.2, require significant modifications to the operation of GSM, which affects all the entities in the network and the deployed phones. This again makes them very unlikely to be deployed. This observation has motivated work on possible changes to the operation of the mobile systems to defeat IMSI catchers that do not require significant changes to the existing network infrastructure and that have minimal overhead. This is the main focus of Chapter 6.

Chapter 5

Retrofitting Mutual Authentication to GSM

5.1 Introduction

In this chapter we describe a novel scheme to add network-to-phone authentication to the GSM mobile system, in a way that is completely transparent to the existing network infrastructure. As described in the previous chapter, GSM only supports authentication of the phone to the network, leaving the system open to a wide range of threats. Although 3G and later generation mobile systems rectify this problem by incorporating mutual authentication between phone and network, GSM remains of huge practical importance worldwide, as discussed in Chapter 1. Therefore, finding ways of upgrading GSM post-deployment appears to be worthwhile. While adding support for mutual authentication in GSM would be highly beneficial, changing the way GSM serving networks operate is not practical.

The scheme described in this chapter introduces a novel modification to the relationship between a SIM and its home network which allows mutual authentication without changing the existing mobile infrastructure, including the phones; the only necessary changes are to the authentication centres and the SIMs. This enhancement, which could be deployed piecemeal in a completely transparent way, not only addresses a number of serious vulnerabilities in GSM but is also the first proposal explicitly designed to enhance GSM authentication that could be deployed without modifying the existing network infrastructure. The main content of this chapter was presented at STM 2016, and has been published in the proceedings [111].

The remainder of the chapter is structured as follows. The adversary model for the scheme is described in Section 5.2. This is followed in Section 5.3 by an introduction

to the notion of *RAND hijacking*. In Section 5.4, the novel enhanced version of the GSM authentication scheme is described, and Section 5.5 describes how the SIM can use the results of network authentication to affect MS behaviour. In Section 5.6, we describe possible changes to the USIM application to support network authentication when a 3G or 4G subscriber roams in a 2G network. An analysis of the novel scheme is provided in Section 5.7. In Section 5.8, we present a ProVerif-based formal verification of the proposed scheme with the goal of verifying the claimed security and privacy properties. The relationship of the proposed scheme to the prior art is discussed in Section 5.9. Finally, the chapter concludes with a summary in Section 5.10.

5.2 Adversary Model

In this section we describe the adversary model for the novel scheme. The scheme is designed to address real-life threats to a GSM network; more specifically, it addresses the attacks described in Sections 4.5.1, 4.5.2, and 4.5.4. In these attacks, the primary goal of the adversary is to force MSs to attach to its fake base station, thereby binding them to a less secure GSM network and exposing them to the attacks described in the previous chapter. The scheme is thus designed to combat active adversaries, as described in the adversary model of Section 4.3.

In designing the scheme we make the underlying assumption that the 3G authentication functions, $f1$ and $f5$, are sound. We also implicitly assume that the SIM, the ME and the network have not been compromised.

The adversary is assumed to be able to:

- run an independent GSM base station;
- maintain connections with a genuine network;
- intercept messages sent across the air interface by a target subscriber;
- modify these air interface messages;
- reply air interface messages; and
- initiate a connection with a target subscriber.

However, the adversary has no control over the core network entities involved in the communication, and is unable to intercept any message transmitted across the core network. Thus, the model does not consider state sponsored adversaries who may be running their own legal serving network.

5.3 RAND Hijacking

We use the term *RAND hijacking* to refer to the idea of using the *RAND*, sent from the network to the UE during AKA, as a way of conveying information from the AuC to the SIM. That is, instead of generating the *RAND* at random, it is generated to contain certain information; this information is sent in encrypted form so that to an eavesdropper the *RAND* is indistinguishable from a random value.

This idea was apparently first described in a patent due to Dupré [79]. However, the use Dupré makes of the idea is rather different to that proposed here. Later, Vodafone introduced the concept of a *special RAND* [173] in 3GPP TSG document S3-030463. The *special RAND* allows the HN to indicate which encryption algorithm is to be used by an ME depending on the attached serving network. Thus, the purpose of the *special RAND* was completely different to that proposed here. The other published references to the notion [71, 110, 163] independently propose the use of RAND hijacking for improving the privacy properties of GSM, 3G and 4G networks. As far as we are aware, no previous authors have proposed the use of this technique with the explicit goal of providing mutual authentication in GSM networks.

5.4 Network-to-SIM Authentication

We now propose a way of using RAND hijacking to enable authentication of the network to the SIM. For this to operate, the SIM must be programmed to support the scheme, as well as possess certain (modest) additional data, as detailed below. The AuC of the network issuing the 'special' SIM must also store certain additional data items for each such SIM, and must generate *RAND* values in a special way for such SIMs. No other changes to existing systems are required. It is important to note that the system could be deployed gradually, e.g. by including the additional functionality in all newly issued SIMs, whilst existing SIMs continue to function as at present.

5.4.1 Prerequisites

In addition to sharing K, $A3$ and $A8$ (as required for executing the standard GSM AKA protocol), the SIM and AuC must both be equipped with the following information and functions:

- functions $f1$ and $f5$, where $f1$ is a MAC function and $f5$ is a cipher mask generation function, both capable of generating a 64-bit output;
- a secret key K_a to be used with functions $f1$ and $f5$ which should be distinct

from K — to minimise memory requirements, K_a and K could, for example, both be derived from a single SIM-specific master key;

- a 48-bit counter to be used to generate and verify sequence numbers[1].

The functions could be precisely the same as their counterparts used in 3G (UMTS). Indeed, the function names and string lengths have deliberately been made identical to those used in 3G systems to make implementation and migration as simple as possible.

5.4.2 Protocol Operation

The novel AKA protocol only differs from the 'standard' GSM AKA protocol (as described in Section 2.4) in the way the authentication vector is generated in the HN, and in the way *RAND* is processed by the SIM; more specifically the scheme changes the way in which *RAND* is calculated by the AuC of the HN, and includes additional steps in processing the received *RAND* by the SIM. Thus, since the scheme involves changes only in the AuC and SIM, it should be clear that the scheme is inherently transparent to the serving network and the ME. We describe below how the AuC and the SIM are changed.

5.4.2.1 Modifications to AuC

The AuC is modified to generate the value *RAND* in a different way; that is, the scheme makes changes to step 1 of the AV generating process described in Section 2.4. To generate a new authentication triple, the AuC proceeds as follows (see Figure 5.1, in which the dotted block represents the usual operation of the AuC).

1. The AuC uses its counter value to generate a 48-bit sequence number *SQN*, which must be a fresh value for this user account.

2. A 16-bit value *AMF* is also generated, which could be set to all zeros or could be used for purposes analogous to the *AMF* value for 3G networks.

3. A 64-bit tag value *MAC* is generated using function $f1$, where

$$MAC = f1_{K_a}(AMF||SQN).$$

4. A 64-bit encrypting mask AK is generated using function $f5$, where

$$AK = f5_{K_a}(MAC).$$

[1] As in 3G, an AuC might choose to manage a single counter shared by all user accounts (see, for example, [129]).

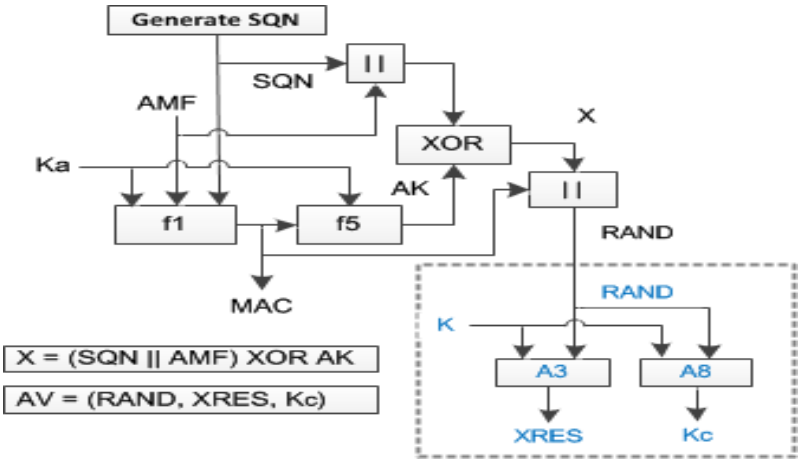

Figure 5.1: Generating an AV in the novel scheme

5. The 128-bit $RAND$ is computed as

$$RAND = ((AMF||SQN) \oplus AK)||MAC.$$

6. The $XRES$ and K_c values are computed in the standard way, that is $XRES = \text{A3}_K(RAND)$ and $K_c = \text{A8}_K(RAND)$.

5.4.2.2 Modifications to SIM

The SIM is modified to allow verification of the network during the operation of AKA; the scheme involves the following changed operation of step 2 of the challenge-response procedure, described in Section 2.4 (see Figure 5.2, in which the dotted block represents the usual operation of the SIM).

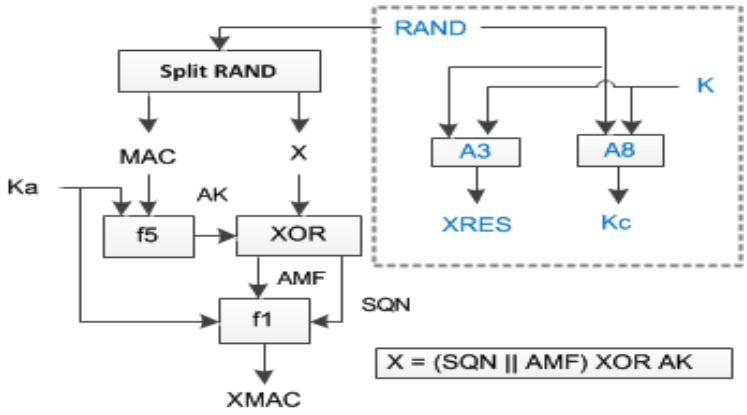

Figure 5.2: Computations at SIM in the novel scheme

1. On receipt of the 128-bit $RAND$ value, the SIM first splits it into two 64-bit strings X and $MAC*$, where $X||MAC* = RAND$.

2. A 64-bit decrypting mask $AK*$ is generated using function $f5$, where

$$AK* = f5_{K_a}(MAC*).$$

3. A 16-bit string $AMF*$ and a 48-bit string $SQN*$ are computed as:

$$AMF*||SQN* = X \oplus AK*.$$

4. A 64-bit tag $XMAC$ is computed as:

$$XMAC = f1_{K_a}(AMF*||SQN*).$$

5. The recovered sequence number $SQN*$ is compared with the SIM's stored counter value, and $XMAC$ is compared with $MAC*$:

   (a) if $SQN*$ is greater than the current counter value **and** $XMAC = MAC*$, then:

      i. the network is deemed to be successfully authenticated;

      ii. the SIM's counter value is updated to equal $SQN*$; and

      iii. $SRES$ and K_c are computed as specified in step 2 of the GSM AKA challenge-response procedure described in Section 2.4;

   (b) if either of the above checks fail, then:

      i. network authentication is deemed to have failed;

      ii. the SIM's counter value is unchanged; and

      iii. $SRES$ and K_c are set to random values.

It should be clear that $AK*$, $AMF*$, $MAC*$ and $SQN*$ should respectively equal the AK, AMF, MAC and SQN values originally computed by the AuC in Section 5.4.2.1.

5.4.3 Design Rationale

The composition of the $RAND$ value in the above scheme has been made as similar as possible to the 128-bit value $AUTN$ used to provide network-to-UE authentication in the 3G AKA protocol. This is for two main reasons. Firstly, as stated above, by adopting this approach it is hoped that implementation of, and migration to, this new

scheme will be made as simple as possible for network operators. Secondly, the 3G AKA protocol is widely trusted to provide authentication, and it is hoped that trust in the novel scheme will be maximised by adopting the same approach.

The only differences between the 3G $AUTN$ and the above construction of $RAND$ are relatively minor, and are as follows.

- In 3G, the AK value is computed as a function of the entire $RAND$, whereas here it is necessarily only computed as a function of the last 64 bits of $RAND$. However, these last 64 bits are computed as a function of data which changes for every authentication triple, and hence the AK should still do an effective job of concealing the content it is used to mask.

- In 3G the AK is only 48 bits long, and is only used to encrypt (mask) SQN. Here we use it to mask SQN and AMF, to ensure that a 'new style' $RAND$ is indistinguishable from an 'old style' randomly generated $RAND$ to any party without the key K_a.

- In 3G, the MAC is computed as a function of $RAND$, SQN and AMF, whereas in the above scheme it is computed only as a function of SQN and AMF, again for obvious reasons. This is the only significant difference from the perspective of authenticating the network to a UE, but we argue below in Section 5.7.2 that this change does not affect the security of the protocol.

The $AUTN$ checking process proposed here and that used in 3G are essentially the same.

One other issue that merits mention is the fact that it is proposed that the SIM outputs random values if authentication fails. It is necessary for the SIM to output values of some kind, since this is part of the existing SIM-ME protocol. That is, placeholder values are required. It is also important for reasons discussed below that the SIM should *not* output the correct session key K_c. The only other 'obvious' placeholder values would be to use fixed strings, but the use of random values seems advantageous if these values are sent across the network (in the case of the $SRES$ value) or used for encryption purposes (for K_c). The advantage of this approach is that it will prevent an eavesdropper distinguishing between a successful and a failure AKA; it will also avoid the leaking of user-specific information and thereby prevent the user linkability attack described in Section 4.5.4.

5.5 Using the Authentication Results

In the previous section we showed how the SIM can authenticate the network; that is, as a result of the modifications to the SIM described in Section 5.4.2.2, the SIM will know whether or not the *RAND* genuinely originates from the AuC and is fresh. However, we did not describe any way for the ME to know whether authentication has failed or succeeded — indeed, the ME will not understand the concept, as we are assuming it is a 'standard' GSM device.

We propose that the SIM application toolkit feature, described in Section 2.6.3, be used to achieve the desired objective. We implicitly assume that the ME supports the *class e* proactive commands. Next, we briefly describe the proactive commands that could be used for this purpose. Although we use 2G terminology to describe the commands, the commands also work with the USIM application. Further details can be found in technical specifications GSM 11.14 [13], 3GPP TS 51.014 [9], and 3GPP TS 31.111 [29].

- *GET CHANNEL STATUS*: This proactive command requests the ME to return the current status of all available data channels. The ME then sends a *TERMINAL RESPONSE* command, containing a channel status data object for each dedicated channel identifier.

- *CLOSE CHANNEL*: This proactive command requests the ME to close the data channel identified by the channel identifier. On receiving the command, the ME releases the data transfer, discards the remaining data, and informs the SIM that the command has been successfully executed using a *TERMINAL RESPONSE* command.

In the event of a network authentication failure, when sending the *SRES* and K_c (in this case random) values back to the ME, the SIM should signal to the phone that it has information to send. When, as a result, the ME sends the *FETCH* command to the SIM, the SIM should respond with the *GET CHANNEL STATUS* command to retrieve details of the channels established in the current connection. Upon receiving the channel information in the *TERMINAL RESPONSE* command, the SIM uses the response status byte in its response to request the ME to send a further *FETCH* command. Once it receives the *FETCH* command, the SIM responds with a *CLOSE CHANNEL* command, specifying the channel identifier of the data channel that has just been established, details of which it received from the ME in response to its previous *CHANNEL STATUS* command. The interactions between a SIM and an ME are summarised in Figure 5.3. The proactive commands issued by the SIM should cause

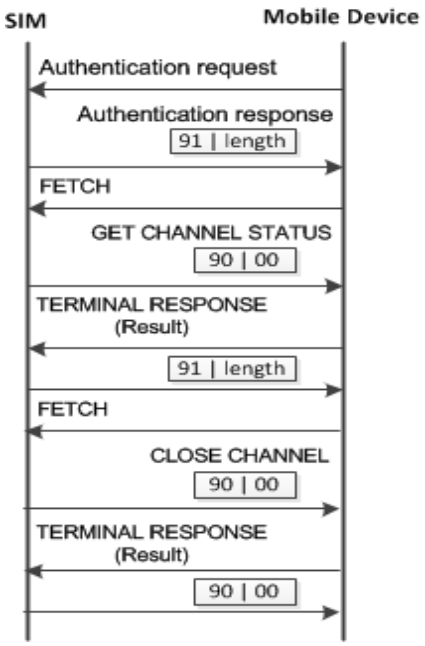

Figure 5.3: SIM-ME interactions to drop any established connection

the phone to drop the connection, and (hopefully) prevent any attempted use of the key K_c. The values 90 and 91, shown in Figure 5.3, represent the values of the *status byte* sent by the SIM in response to the previous command, where the value 90 means *OK*, and the value 91 instructs the ME to issue a *FETCH* command to retrieve data from the SIM. The 'length' with the status byte 91 indicates the length of the data in bytes which the SIM wants to send.

5.6 Inter-Networking Issues

We next describe possible changes to the USIM application to support network authentication in the case where a 3G or 4G subscriber roams in a 2G network.

As discussed in Section 3.5.1 a UE never runs GSM AKA when it roams in a 2G network; instead the USIM operates in virtual 2G mode to run the GSM authentication protocol. Hence, the network authentication logic introduced in the scheme proposed in Section 5.4 needs to be incorporated into the operation of virtual 2G mode to protect 3G and 4G subscribers roaming in 2G networks. We describe the modified virtual 2G mode below.

In virtual 2G mode, the USIM receives *RAND* as the authentication parameter. On receiving *RAND*, the USIM first verifies its integrity and authenticity. To do so the USIM follows the steps described in Section 5.4.2.2, except for step 5(a)iii. Step 5(a)iii is

replaced with the usual operation of the virtual 2G mode, in which the USIM computes the 2G *SRES* and K_c, using the functions $f2$, $f3$, $f4$, $c2$, and $c3$ as described in Section 3.5.1. The USIM uses the authentication result as described in Section 5.5.

The steps introduced into the operation of the modified virtual 2G mode involve the use of functions $f1$ and $f5$. However, as described in Section 5.4.1, these functions are assumed to be available in the USIM. It is also assumed that the USIM has access to the key K_a.

5.7 Analysis

5.7.1 Deployment Issues

We next consider certain practical issues that may arise when using the scheme proposed in Section 5.4.

It seems that at least some GSM networks issue authentication triples in batches (see Section 3.3.1 of GSM 03.20 [2]), thereby reducing the inter-network communications overhead. Currently, the order in which GSM authentication triples are used does not matter. However, in the scheme described above, triples must be used in a way that ensures that the embedded *SQN* is fresh. This may seem problematic; however, since the requirement to use authentication datasets in the correct order already applies to the corresponding 5-tuples used in 3G, serving networks will almost certainly already be equipped to do this.

In existing GSM networks it is possible, although prohibited by the technical specifications [2], for serving networks to 're-use' authentication triples, i.e. to send the same *RAND* value to a MS on multiple occasions. This will no longer work with the new scheme, since the SIM will detect re-use of a *RAND* value. Arguably this is good, since re-use of *RAND* values is highly insecure: such behaviour would allow the interceptor of a *RAND*/*SRES* pair to impersonate a valid MS and perhaps steal service at that MS's expense, an attack that would be particularly effective in networks not enabling encryption.

Finally note that, in order to fully implement the scheme described in Section 5.4, MEs need to support *class e* STK commands, although, as discussed in Section 2.6.3, this proportion seems likely to be very high. It is not clear what proportion of mobile phones in current use support these STK commands.

5.7.2 Security

We divide our security discussion into three parts: confidentiality and privacy issues, authentication of network to SIM, and authentication of SIM to network.

5.7.2.1 Confidentiality and Privacy Issues

In 'standard' GSM the $RAND$ value is randomly selected, and so does not reveal anything about the identity of the phone to which it is sent. In the scheme proposed in Section 5.4, the $RAND$ is a function of a SIM-specific key as well as a potentially SIM-specific SQN value. However, the SQN is sent encrypted, and, assuming the functions $f1$ and $f5$ are well-designed, an interceptor will not be able to distinguish an intercepted $RAND$ computed according to the new scheme from a random value. Thus the scheme does not introduce a new threat to identity confidentiality.

The new scheme does not change the way the data confidentiality key K_c is generated, so the strength of data confidentiality is not affected.

5.7.2.2 Network-to-SIM Authentication

The novel protocol for network-to-SIM authentication bears strong similarities to the corresponding protocol for 3G. It also conforms to the one-pass unilateral authentication mechanism specified in clause 5.1.1 of ISO/IEC 9798-4 [103, 105]. All the protocols in this standard have been formally analysed (and shown to be secure) by Basin, Cremers and Meier [54, 55]. Whilst these arguments do not provide a completely watertight argument for the protocol's security, it is clearly a significant improvement over no authentication at all.

An interesting side observation deriving from the novel scheme is that the 3G and 4G AKA protocols appear to be overly complex. The randomly generated $RAND$ value sent from the network to the SIM, which is used to authenticate the response from the SIM to the network, is actually unnecessary, and the $AUTN$ value could be used in exactly the same way as the $RAND$ is currently. Whilst such a change is not possible in practice, it would have avoided the need for the AuC to generate random values and saved the need to send 16 bytes in the AKA protocol.

It is interesting to speculate why this design redundancy is present. It seems possible that network-to-SIM authentication was added as a completely separate protocol to complement the GSM-type SIM-to-network authentication mechanism, and no-one thought how the two mechanisms could be combined and simplified (as in the mechanism we propose).

5.7.2.3 SIM-to-Network Authentication

The novel scheme does not affect how the existing SIM-to-network authentication protocol operates, except that a random $RAND$ is replaced by one which is a cryptographic function of a sequence number. The new-style $RAND$ remains unpredictable to anyone not equipped with the key K_a, and is deterministically guaranteed to be non-repeating (a property that only holds in a probabilistic way for a random $RAND$). To see why the $RAND$ is non-repeating, suppose two separate $RAND$ values sent to the same USIM incorporate the same MAC values (as necessary if they are to be the same). It follows that the AK values used to mask the $SQNs$ embedded in the $RAND$ values will also be the same and thus, since the SQN values themselves will be different, the two $RAND$ values will also differ. That is, it possesses precisely the qualities required by the existing protocol, and hence the security of SIM-to-network authentication is unaffected. We give a formal proof of our claim below.

Claim 5.7.1. *The value of RAND in the new scheme is deterministically guaranteed to be non-repeating.*

Proof. Suppose R_a and R_b are $RAND$ values generated by an AuC for a subscriber, where

- $R_a = ((SQN_a \parallel AMF) \oplus AK_a) \parallel MAC_a$ and
- $R_b = ((SQN_b \parallel AMF) \oplus AK_b) \parallel MAC_b$.

and where SQN_a and SQN_b are distinct values of SQN.

It is sufficient to prove that the value R_a and R_b must differ if SQN_a and SQN_b differ. For simplicity we assume that the value of AMF is constant, which is generally true for real-world networks.

Suppose $R_a = R_b$. Then $MAC_a = MAC_b$, since R includes MAC. Hence $AK_a = AK_b$, since $AK = f5_{K_a}(MAC)$. It follows that $R_a = (SQN_a \parallel AMF) \oplus AK_a \neq (SQN_b \parallel AMF) \oplus AK_b = R_b$. The result follows by contradiction.

□

5.7.3 Impact on Known Attacks

We conclude our analysis of the protocol by considering how it affects possible attacks on GSM networks.

5.7.3.1 Fake Network Attacks

As discussed in Section 4.5.1, if a phone joins a fake GSM serving network, then this fake network can send any *RAND* value it likes as part of the AKA protocol, and the MS will complete the process successfully. If the network does not enable encryption, then communications between the MS and the network will work correctly, which could enable the network to act as an eavesdropping man-in-the-middle by routing calls from the captured MS via a genuine network. This will no longer be true if the new scheme is implemented, since the SIM will instruct the ME to drop the connection when supplied with a non-genuine *RAND* value.

Of course, it may be possible for a fake network to avoid the AKA protocol altogether, and simply start communication with a newly attached MS. However, it is not clear whether MEs will accept such unauthenticated communication.

5.7.3.2 Barkan-Biham-Keller Attacks

We next consider a particular type of fake network attack, namely the Barkan-Biham-Keller attack described in Section 4.5.2. The attack requires the re-sending of an 'old' *RAND* to an MS. Since in the new scheme the SIM verifies the sequence number embedded in the authentication challenge as part of the enhanced AKA, the new scheme will clearly prevent such an attack, i.e. the Barkan-Biham-Keller attack will be prevented, at least in most practical scenarios.

5.7.3.3 User Linkability Attacks

As discussed in Section 4.5.4, this attack requires the re-sending of an 'old' *RAND* to an MS, and makes use of the SIM-computed *SRES* value for that specific *RAND*. Since the new scheme returns a random *SRES* when the same *RAND* is re-sent to an MS, the new scheme will prevent such an attack.

5.8 Formal Verification

To verify the security properties of the proposed scheme formally we make use of *ProVerif*, an automatic cryptographic protocol verifier [58]. We first introduce the ProVerif tool, following Blanchet, Smyth and Cheval [60]. We then give a ProVerif model of the proposed scheme, together with a formalisation of the security and privacy properties under analysis. The formal model described here only considers the 2G SIM, and not the USIM. Finally, we describe the results of the ProVerif analysis.

5.8.1 The ProVerif Tool

The ProVerif tool is designed to be used to verify the secrecy and authentication properties of a cryptographic protocol that interacts using public communication channels. Since these channels are assumed to be controlled by a very powerful environment, capturing an attacker with Dolev-Yao capabilities [78], ProVerif simulates the Dolev-Yao attacker model. In this model an attacker has complete control over the communication channels; that is, the attacker may read, modify, delete, and inject messages. However, cryptography is assumed to be perfect; that is, the attacker is only able to perform cryptographic operations when in possession of the required keys. In other words, it is restricted to applying only the cryptographic primitives specified by the user. The environment also captures the behaviour of dishonest participants; hence, only honest participants need to be modelled in ProVerif. The tool can analyse protocols with unbounded sessions using automated procedures.

To analyse a protocol in ProVerif, a ProVerif model of the protocol and the security properties of interest are provided as input to the tool. The tool translates the input into a set of statements, known as *Horn clauses*, which are automatically provided to the tool's *resolution algorithm*. The resolution algorithm verifies the security properties using these Horn clauses, and outputs an indication of whether the security properties hold for the input protocol model. If the tool can derive a fact in contradiction to a desired security property, it finds an attack and outputs the actions an attacker may take to break the security property; such an attack trace can be used to reconstruct the attack. However, when no fact contradicting the desired security property can be derived, the tool outputs that the security property holds.

In certain cases the tool may generate a *false attack* or may not terminate an analysis. Derivation of facts during the translation into Horn clauses, or an infinite loop generated by the Horn clauses, might cause such an outcome. However, such outcomes are unusual in practice. Although the analysis by ProVerif is not complete, which means that it may not be capable of proving a property that holds, it is sound; that is, when ProVerif verifies that a property is satisfied, then the model does guarantee that property.

ProVerif has been used to verify a wide range of cryptographic protocols. For example, Smyth et al. [149] and Kremer, Ryan and Smyth [115] used it to analyse the security properties of a number of e-voting protocols, and Delaune, Kremer and Ryan [75] used it to analyse the privacy properties of a similar set of protocols. Chen and Ryan [69] used ProVerif to evaluate the authentication protocols employed by the *trusted platform module* (TPM), and discovered vulnerabilities. Abadi and Blanchet [43] used ProVerif to verify the certified email protocol [44]. ProVerif has also been used to verify the se-

curity and privacy properties of the 2G, 3G and 4G mobile networks. Tang, Naumann and Wetzel [156] analysed the AKA protocols in inter-generation mobile systems using ProVerif. Arapinis et al. [48, 49] and van den Broek, Verdult and de Ruiter [163] used ProVerif to verify the security properties in their 'fixed' version of the protocol.

5.8.2 Formal Model of the New Scheme

ProVerif's input language is the *process calculus*, used to model concurrent systems; hence, protocols are modelled using process calculus syntax. We next describe the part of the language we use to model the scheme proposed earlier in this chapter. Further details of the grammar of the ProVerif language and its syntax can be found in the ProVerif manual [60].

A ProVerif model of a protocol is divided into three parts: the *declarations*, the *process macros*, and the *main process*. We next provide a summary of these three parts of the model for the enhanced GSM AKA scheme – full details are provided in Appendix A.1. The ProVerif model we present immediately below is a modified version of the model due to Tang, Naumann and Wetzel [154, 156].

5.8.2.1 The Declarations

The declarations part includes a finite set of *types* and *free variables*, and formalises the behaviour of cryptographic primitives using a set of functions known as *constructors*, and corresponding rewrite rules known as *destructors*. Constructors are used to build *terms* used by a protocol, and take the form *fun* $f(t_1, ..., t_n) : t$, where f is a constructor, t is its return type, and $t_1, ..., t_n$ are the types of its arguments. In the syntax discussed here, n is always a positive integer. The term returned by a constructor can be a single variable. Destructors are used to manipulate terms formed by constructors, and take the form *reduc forall* $x_1 : t_1, ..., x_n : t_n; g(M_1, ..., M_n) = M$, where g is a destructor, and the terms $M_1, ..., M_n$ and M are built from the application of constructors to variables $x_1, ..., x_n$ of types $t_1, ..., t_n$, respectively. When an instance of the term $g(M_1, ..., M_n)$ is encountered during execution, it is replaced by M.

Listing 5.1 contains the salient parts of the declarations for the enhanced GSM AKA protocol. The full listing is given in Appendix A.1. In Listing 5.1, the $a3$, $a8$, $f1$, $f5$ constructors model the authentication-specific cryptographic functions, *bitstring* is the ProVerif's built-in type, and *mac*, *key*, *resp*, *sessionKey*, and *anonymityKey* are user-defined types. The constructor *encrypt* models the $\oplus$ operator used to conceal the *SQN* value in the proposed scheme. To retrieve the *SQN* value from *RAND* in the MS, the destructor *decrypt* is used.

Listing 5.1: Enhanced GSM AKA model: Summary of declarations

```
(* Constructors and destructors *)
  fun a3(bitstring, mac, key): resp.
  fun a8(bitstring, mac, key): sessionKey.
  fun f1(bitstring, key): mac.
  fun f5(mac, key): anonymityKey.
  fun encrypt(bitstring, anonymityKey): bitstring.

  reduc forall m: bitstring, k: anonymityKey; decrypt(encrypt(m, k),
      k) = m.

(* Secrecy query *)
  free sqn: bitstring [private].
  query attacker(sqn).

(* Authentication query *)
  query x1: ident, x2: sessionKey; event(endSN(x1, x2)) ==> event(
      begSN(x1, x2)).
  query x1: ident, x2: sessionKey; event(endMS(x1, x2)) ==> event(
      begMS(x1, x2)).
```

The declarations part also formalises the security properties to be verified. We are interested in verifying the secrecy property of *SQN* and the mutual authentication feature. The ProVerif tool verifies the secrecy of any term by proving the *reachability* property, and the authentication features are verified using *correspondence assertions* [59]. The following *query* syntax is used to achieve our goal.

- *query attacker (M)*, queries the secrecy of the term M. If an attacker finds a way to learn M, the query fails. In other words, ProVerif attempts to verify that any state in which the term M is known to the adversary is unreachable.

- *query* $x_1 : t_1, ..., x_n : t_n$; $event(e(M_1, ..., M_j)) \implies event(e^*(N_1, ..., N_k))$, where $M_1, ..., M_j, N_1, ..., N_k$ are terms constructed using the variables $x_1, ..., x_n$ of types $t_1, ..., t_n$ and e, e^* are declared *events*, queries a possible relationship between events. The events mark important stages reached by the protocol but do not otherwise affect the behaviour of the process, and the relationships between events are specified as correspondence assertions [174]. The query is satisfied if, for each occurrence of the event e, there is a previous execution of event e^*. Moreover, the parameterisation of the events e and e^* must satisfy any relationships defined by its arguments; that is, the variables $x_1, ..., x_n$ have the same value in $M_1, ..., M_j$ and in $N_1, ..., N_k$.

In line 11 of Listing 5.1, we declare a *free* variable *sqn* of type *bitstring*. Free variables are available to the attacker unless they are declared private by appending [*private*]. Since we are interested in verifying the secrecy of this variable, we declare the variable as private. Lines 12, 15, and 16 in Listing 5.1 query the security properties of interest for the proposed scheme. The events *begSN*, *endSN*, *begMS*, *endMS* are described below, in the process description where they are used.

5.8.2.2 The Process Macros

Process *macros* are sub-processes defined in order to ease development. Macros take the form *let* $R(x_1 : t_1, ..., x_n : t_n) = P$, where R is the macro name, P is the sub-process being defined, and $x_1, ..., x_n$ of types $t_1, ..., t_n$ respectively are the free variables of P. We define three process macros to model the processes of the MS, SN, and HN.

We assume that the MS and SN communicate with each other through a public channel (the *pubChannel* variable in the model) and that the communication channel between the SN and HN (the *secureChannel* variable in the model) is private. The receipt of a message in a process is represented by $in(c, x)$, where c is the communication channel and x is the received message. Similarly, sending a message is represented by $out(c, y)$, where c is the communication channel and y is the sent message. A destructor application of the form *let* $M = D$ in P else Q tries to rewrite D and matches the result with M; if this succeeds, then the variables in M are instantiated accordingly and P is executed; otherwise, Q is executed. The conditional construct, *if* $M = N$ *then* P *else* Q, checks the equality of two terms M and N, and then behaves as P or Q accordingly. We omit the *else* branch of a let or a conditional, when the process Q is 0, meaning a null process.

The MS process, described in the *processMS* sub-process (see Listing 5.2), sends the IMSI across the public channel for authentication, and then waits for an authentication challenge (*RAND*), which is the concatenation of the encrypted *SQN* and the MAC. On receiving the *RAND*, the process uses the destructor functions to retrieve *SQN*, to compute a MAC, and to compare the computed MAC with the received MAC. If they match, authentication of the SN is validated; in such a case, the process marks the event *endMS* (line 9 in Listing 5.2) with the IMSI and the corresponding session key as event parameters. The process also marks the event *beginSN* (line 10 in Listing 5.2) with the IMSI and the corresponding session key as event parameters and sends the computed *SRES*, the basis of MS authentication, across the public channel to be received by the SN process. However, if the MAC comparison fails, the process sends a random value as *SRES* across the public channel. The event *endMS* indicates that authentication of the SN by the MS has completed. Similarly, the event *begSN* implies that authentication

of the MS by the SN has started.

Listing 5.2: Enhanced GSM AKA model: MS process (highlights)

```
1  let processMS=
2    out(pubChannel, (ID, imsi_ms));
3    in(pubChannel, (=CHALLENGE, enc_sqn_ms: bitstring, mac_ms: mac));
4    let ak_ms: anonymityKey = f5(mac_ms, ki) in
5    let sqn_ms: bitstring = decrypt(enc_sqn_ms, ak_ms) in
6    if f1(sqn_ms, ki) = mac_ms then (
7          let res_ms: resp = a3(enc_sqn_ms, mac_ms, ki) in
8          let kc_ms: sessionKey = a8(enc_sqn_ms, mac_ms, ki) in
9          event endMS(imsi_ms, kc_ms);
10         event begSN(imsi_ms, kc_ms);
11         out(pubChannel, (SRES, res_ms))) else (
12   new d_res: resp;
13   out(pubChannel, (SRES, d_res))).
```

SN protocol execution is described in the *processSN* sub-process (see Listing 5.3). On receiving the IMSI from the public channel, the process sends the IMSI across the private channel to the HN, and waits for an AV. When it receives the AV, it marks the event *begMS* (line 6 in Listing 5.3), with the IMSI and the corresponding session key as event parameters, and sends the received authentication challenge across the public channel to the MS. The process then waits for the *SRES*, and on receiving a value from the public channel, it compares it with the *XRES* in the AV. If they match then authentication of the MS is completed; in such a case, the process marks the event *endSN* (line 10 in Listing 5.3), with the IMSI and corresponding session key as event parameter. Like the events described in *processMS*, the event *begMS* implies that authentication of the SN by the MS has started, and the event *endSN* indicates that authentication of the MS by the SN has completed.

HN protocol execution is described in the *processHN* sub-process (see Listing 5.4). This process always interacts with the secure channel. It receives an AV request from the SN, computes an AV using defined constructors and destructors, and sends the AV to the SN.

5.8.2.3 The Main Process

The main process of the ProVerif model encodes the complete protocol. We encode the modified GSM AKA protocol using the macros defined in the previous section. We model the execution of unbounded number of MS processes as *(!processMS)*, where the symbol '!' represents replication and instantiates the parallel execution of an unbounded

Listing 5.3: Enhanced GSM AKA model: SN process (highlights)

```
let processSN=
  in(pubChannel, (=ID, imsi_sn: ident));
  out(secureChannel, (AV_REQ, imsi_sn));
  in(secureChannel, (=AV, imsi_hn_sn: ident, enc_sqn_sn: bitstring,
  mac_sn: mac, xres_sn: resp, kc_sn: sessionKey));
  event begMS(imsi_hn_sn, kc_sn);
  out(pubChannel, (CHALLENGE, enc_sqn_sn, mac_sn));
  in(pubChannel, (=SRES, res_sn: resp));
  if res_sn = xres_sn then
        event endSN(imsi_hn_sn, kc_sn).
```

Listing 5.4: Enhanced GSM AKA model: HN process (highlights)

```
let processHN=
  in(secureChannel, (=AV_REQ, imsi_hn: ident));
  get keys(=imsi_hn, ki_hn) in
  let mac_hn: mac = f1(sqn, ki_hn) in
  let ak_hn: anonymityKey = f5(mac_hn, ki_hn) in
  let enc_sqn_hn: bitstring = encrypt(sqn, ak_hn) in
  let xres_hn: resp = a3(enc_sqn_hn, mac_hn, ki_hn) in
  let kc_hn: sessionKey = a8(enc_sqn_hn, mac_hn, ki_hn) in
  out(secureChannel, (AV, imsi_hn, enc_sqn_hn, mac_hn, xres_hn,
     kc_hn)).
```

number of copies of *processMS*, in parallel to the SN and HN processes. The parallel execution of processes P and Q is represented as $P|Q$. Listing 5.5 shows the main process that embodies the modified GSM AKA protocol. A full listing of the ProVerif code can be found in Appendix A.1.

Listing 5.5: Enhanced GSM AKA model: Main process

```
  process
        ((!processMS) | processSN | processHN)
```

5.8.3 Verification Result

We ran the encoded protocol described in Section 5.8.2 in ProVerif to verify the secrecy and authenticity properties. The tool output *RESULT not attacker(sqn[]) is true*,

which means that the attacker is not able to get the value of SQN. Since AMF is transferred in exactly the same way as SQN and is used similarly in the proposed scheme, in the protocol model we do not model AMF explicitly (SQN in the model can be thought of as the concatenation of SQN and AMF); hence the ProVerif verification also shows that AMF secrecy is maintained.

In response to the first correspondence assertion query, the tool output *RESULT event(endSN*$(x1_{1029}, x2_{1030})$ $\implies$ *event(begSN*$(x1_{1029}, x2_{1030})$*) is true*, which indicates that authentication of the MS by the SN is achieved. In response to the second correspondence assertion query, ProVerif returned *RESULT event(endMS(x1,x2))* $\implies$ *event(begMS(x1,x2)) is true*, which implies that network authentication by the phone is achieved. These two results imply that the mutual authentication property holds.

Table 5.1: Comparison of security properties

Properties/Protocol	GSM AKA	Proposed scheme
Authentication of Phone	Yes	Yes
Authentication of Network	No	Yes
Privacy of SQN	NA	Yes
Privacy of AMF	NA	Yes

We also formally modelled the existing GSM AKA protocol. The ProVerif model of GSM AKA that we used is presented in Appendix A.2. We ran the encoded protocol in ProVerif to confirm that the existing GSM AKA lacks the network authentication property. Table 5.1 compares the security properties of the novel scheme with the existing GSM AKA, where a 'Yes' implies that the ProVerif tool has verified that the indicated property holds, and 'NA' means that the indicated property is not applicable. The ProVerif outputs for both the existing GSM AKA protocol and the enhanced version are given in Appendix A.3.

5.9 Relationship to the Prior Art

This is by no means the first practical proposal for enhancing GSM to incorporate mutual authentication. Indeed, the 3G AKA protocol, discussed in Section 3.4.1, can be regarded as doing exactly that. Although two 3GPP TSG documents [81, 82], described in Section 4.6.1, proposed the introduction of network authentication into the GSM network, neither of the schemes was adopted, presumably because of cost/feasibility issues. Other proposals have been made, as discussed in Section 4.6.1. However, all previous proposals are completely impractical in that they would require changes to the GSM infrastructure. Such major changes to a widely deployed scheme are simply

not going to happen.

The most similar proposals to that given here are some of the other schemes using RAND hijacking, summarised in Section 5.3. In particular, van den Broek, Verdult and de Ruiter [163] propose a similar structure for a hijacked GSM *RAND*, in their case including a sequence number, a new temporary identity for the SIM, and a MAC, all encrypted in an unspecified way. However, their objective is not to provide authentication of the network to the SIM, but to provide a way to reliably transport new identities from the AuC to the SIM.

5.10 Summary

In this chapter we have proposed a method for enhancing the GSM AKA protocol to provide authentication of the network to the MS, complementing the MS-to-network authentication already provided. This provides protection against some of the most serious threats to the security of GSM networks. This is achieved in a way which leaves the existing serving network infrastructure unchanged, and also does not require any changes to existing MEs (mobile phones). That is, unlike previous proposals of this general type, it is practically realisable.

We have analysed the proposed modification to GSM AKA using the ProVerif tool, and shown that the modified protocol provides mutual authentication without leaking any confidential data.

A number of practical questions remain to be answered, including the proportion of MEs supporting 'class e' STK commands, the behaviour of MEs in networks which never perform the AKA protocol, and whether SNs can be relied upon to use GSM authentication triples in the intended order. Discovering answers to these questions remains as future work.

Chapter 6

Improving Air Interface User Privacy in GSM

6.1 Introduction

Although a number of possible modifications to 2G protocols to enhance user privacy have been proposed, as described in Section 4.6.2, they all require significant alterations to the existing deployed infrastructures, something that is almost certainly impractical to achieve in practice. In this chapter we introduce new approaches to the use and management of multiple IMSIs in a SIM with the goal of enhancing user pseudonymity on the air interface in a way which does not require any changes to the existing deployed network infrastructures, i.e. to the serving networks or the mobile devices. Similar approaches also form the basis of schemes designed to enhance user identity privacy in 3G and 4G, described in Chapter 9.

The only changes required by the new scheme are to the operation of the authentication centre in the home network and to the SIM, both owned by a single entity in the mobile system. We describe two approaches to the use and management of multiple IMSIs in a SIM, and report on experiments to validate their deployability.

The remainder of the chapter is structured as follows. A description of the threat model is presented in Section 6.2. Section 6.3 outlines a novel approach to improving air interface user privacy using multiple IMSIs. The procedure to update an IMSI in a SIM is described in Section 6.4. Sections 6.5 and 6.6 provide descriptions of two proposed approaches to the use and management of multiple IMSIs in a SIM. Results from our experimental evaluation are presented in Section 6.7. An analysis of the proposed approaches is presented in Section 6.8. Section 6.9 provides a discussion of related work. Finally, conclusions are drawn in Section 6.10.

6.2 Threat Model

In this section we describe the threat model underlying the schemes proposed in this chapter. As described in Section 2.3.1, the IMSI is used to identify the subscriber for authentication and access provision; since the IMSI is a permanent user identity, the air interface protocols are designed to minimise the number of circumstances in which it is sent across the air interface. However, we have already observed that there are circumstances (see Section 4.2.1) in which an adversary can cause an MS to send its IMSI across the radio path. This is the threat we aim to mitigate.

We observe that GSM AKA does not provide network authentication, and so is unable to guarantee an authenticated channel between the MS and the HN. We implicitly assume that the SIM, ME, and the network have not been compromised.

The main objective of the proposed schemes is to reduce the impact of IMSI disclosure, thereby enhancing user privacy. That is, although the possibility of IMSI compromise remains unchanged, we propose making the IMSI a short term identity and hence avoid the disclosure of a single long-term user identity. In doing so we must also ensure that two different IMSIs for the same MS are not linkable, at least via the network protocol. The main risk introduced by use of multiple-IMSI schemes is the possibility of loss of IMSI synchronisation between MS and HN; this issue is addressed in Section 6.8.

As in the adversary model described in Section 5.2, we consider active adversaries, more specifically IMSI catchers, in the threat model.

6.3 A Pseudonymity Approach

The idea underlying the schemes described in this chapter is to associate multiple IMSIs with a single account, thereby supporting a form of pseudonymity on the air interface. The use of multiple IMSIs is described here using GSM terminology; however, a precisely analogous approach would apply equally to both 3G and 4G systems.

At present, a SIM holds one IMSI along with other subscription and network parameters. We propose that a SIM and the HN support the use of varying IMSIs for a single user account, in such a way that no modification is required to the operation of any intermediate entities, notably the serving network and the ME itself. This allows the provision of a more robust form of pseudonymity without making any changes to the air interface protocol. In this section we consider how a change of IMSI can be made.

The following issues need to be addressed to allow use of multiple IMSIs.

- *Transferring IMSIs*: Clearly, before a SIM switches to a new IMSI, it must be present in the SIM and in the database of the HN. Also, new IMSIs must always be chosen by the HN to avoid the same IMSI being assigned to two different SIMs. This requires a direct means of communication between the HN and the SIM (which must be transparent to the SN and the ME, since our objective is to enable changing of an IMSI without making any changes to existing deployed equipments). In Sections 6.5 and 6.6, we introduce two strategies for transferring IMSIs from the HN to a SIM.

- *Triggering an IMSI change*: Whether the SIM or the HN is responsible for triggering a change of IMSI, logic needs to be implemented to cause such a change to take place. Regardless of whether the SIM or the HN makes the decision, logic needs to be in place in the SIM either to make the decision or to receive the instruction to make the change from the HN; for convenience we refer to this logic as an application, although this is not intended to constrain how it is implemented. The decision-making logic could take account of external factors, including, for example, the elapsed time or the number of AKA interactions since the last change; indeed, if the ME included an appropriate user-facing application, then it might also be possible to allow user-initiated changes. Of course, if the HN is responsible for triggering the change of IMSI, then it needs a means of communicating its decision to the SIM that is transparent to the existing infrastructure, including the SN and the ME. This issue is addressed in Sections 6.5 and 6.6.

- *Use of a new IMSI*: Clearly the IMSI needs to be changed in such a way that both the HN and the SIM know at all times which IMSI is being used, and the HN always knows the correspondence between the IMSI being used by the SIM and the user account. An IMSI change can be triggered either by the SIM or by the HN, as we describe above. However, use of a new IMSI is always first implemented by the SIM, since it is the appearance of a mobile device in a network using a particular IMSI which causes a request to be sent by the SN for authentication information for use in the AKA protocol. That is, when the ME sends an IMSI to the SN, it is forwarded to the HN. Once the HN sees the 'new' IMSI it knows that an IMSI change has occurred and can act accordingly.

  This requires that the HN knows that both the previously used IMSI and the 'new' IMSI belong to the same account. This will require some minor changes to the operation of the HN's account database, i.e. to allow more than one IMSI to point to a single account. However, this does not seem likely be a major problem

in practice.

- *Rate of change of IMSI*: The rate of change of IMSI will clearly be decided by the SIM-issuing network (which equips the SIM with the IMSI-changing application). We observe in this context that Section 10.3.2 of GSM 11.11 [12] and 3GPP TS 51.011 [10], and Section 4.2.2 of 3GPP TS 31.102 [28] recommend that IMSI updates should not occur frequently. The rate of change of an IMSI could be determined by the customer contract with the issuing network; for example, a SIM which changes its IMSI frequently might cost more than a fixed-IMSI SIM (or one that only changes its IMSI occasionally), and could be marketed as a special 'high-privacy' service.

- *Implementing an IMSI change*: Since the use of a new IMSI is always implemented by the SIM, a mechanism will be required for the SIM to indicate to the ME that the IMSI has changed; this is important to ensure the use of the new IMSI across the air interface. We propose that the SIM application toolkit feature described in Section 2.6.3 be used by the SIM to achieve this objective. We describe the details of the procedure in Section 6.4 below.

As noted above, using multiple IMSIs requires a direct and transparent means of communication between the HN and the SIM. The *unstructured supplementary service data* (USSD) protocol appears at first sight to be a possible channel for such communications. However, the security properties of USSD are not satisfactory, since they are dependent on the underlying network [62]. As a result we do not consider the use of USSD further.

6.4 Transfer of New IMSI to ME

We now describe the procedure to update an IMSI in the SIM and to allow the ME to be made aware of the new IMSI. We assume that the SIM will change the IMSI when the ME is in idle state; we therefore propose that, as part of the IMSI change process, the SIM initiates a proactive command that enables an exchange of data eventually resulting in the ME learning the new IMSI. The commands to achieve the desired objectives are described below. Further details can be found in technical specifications GSM 11.14 [13], 3GPP TS 51.014 [9], and 3GPP TS 31.111 [29].

- *REFRESH*: This proactive command requests the ME to carry out a SIM initialisation using the procedure described in Section 11.2.1 of GSM 11.11 [12], and/or

advises the ME that the contents of EFs on the SIM have been changed. The command also makes it possible to restart a card session by resetting the SIM.

The command supports five different modes as follows.

- *SIM initialisation*: This mode tells the ME to carry out SIM initialisation using the procedure described in Section 11.2.1 of GSM 11.11 [12], starting after the *card holder verification 1* (CHV1) verification procedure.
- *File change notification*: This mode informs the ME of the identity of the EFs that have been changed in the SIM. If there is an image of certain SIM EFs in the ME memory, this information can be used by the ME to determine whether it needs to update this image.
- *SIM initialisation and file change notification*: This is a combination of the two modes above.
- *SIM initialisation and full file change notification*: This mode causes the ME to perform the SIM initialisation procedure of the first mode above, and also informs the ME that EFs have been changed in the SIM. If there are any images of SIM EFs in the ME memory, the ME shall completely update them.
- *SIM reset*: This mode causes the ME to run the GSM session termination procedure and to deactivate the SIM in accordance with GSM 11.11 [12]. Subsequently, the ME activates the SIM again and starts a new card session. The SIM reset mode is used when a SIM application requires complete SIM initialisation procedures to be performed.

- *STATUS*: This command is frequently issued by an ME to a SIM, and is used to check the presence of the SIM. When the ME is an idle state, the command gives an opportunity for a proactive SIM to indicate that the SIM wants to issue a SIM application toolkit command to the ME.

The following steps are used to implement the IMSI change and to notify the change of IMSI to the ME.

1. As noted in Section 2.6.1, the IMSI is contained in the elementary file EF_{IMSI}. When the SIM wishes to change the IMSI, it first updates this file accordingly.
2. At the first opportunity, the SIM uses the response *status byte* of the *STATUS* command to indicate to the ME that it wishes to issue a command (see Figure 6.1). The value 91 and the *length*, shown in Figure 6.1, represent the value

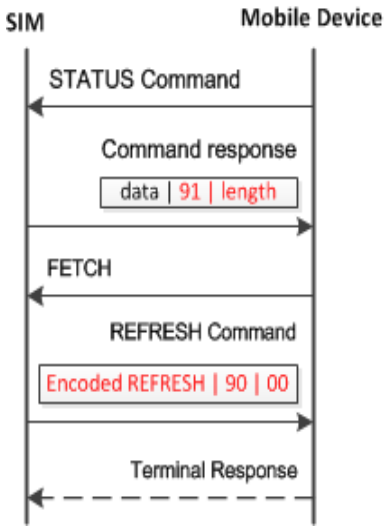

Figure 6.1: SIM-ME interactions to transfer the new IMSI

of the *status bytes* sent by the SIM in response to the *STATUS* command, and instruct the ME to issue a *FETCH* command to retrieve data from the SIM, where the *length* indicates the length of data in bytes which the SIM wants to send (in this case the size of *REFRESH* command, see below).

3. When the ME responds with a *FETCH* command, the SIM sends a *REFRESH* command to the ME and sets the *status bytes* as 90|00, indicating a successful command execution. Precisely which *REFRESH* mode should be used here is up to the operator issuing the SIM; the main requirement is that the command ensures that the ME refreshes its copy of the EF_{IMSI}.

4. On receiving the *REFRESH* command, the ME performs according to the mode set in the command, and informs the SIM about the result of the command execution using a *TERMINAL RESPONSE* command. However, when the command mode is set to *SIM reset*, the ME does not send a *TERMINAL RESPONSE*; instead the SIM application interprets a new activation of the contacts of the SIM as an implicit *TERMINAL RESPONSE*. The *REFRESH* command, if used in an appropriate mode and with suitable parameters, will cause the ME to read the EF_{IMSI}, allowing it to learn the new IMSI. As a result, the next time that the ME needs to send its IMSI to the SN, it will send the new value.

6.5 Predefined Multiple IMSIs

In this and the next section, we describe two general approaches to the practical management of multiple IMSIs for a SIM. Our first means of deploying multiple IMSIs involves a SIM being pre-equipped with a number of IMSIs. These IMSIs are all associated with a single account in the HN's account database. Initially, one of the IMSIs is stored in EF_{IMSI}. We propose below two ways of initiating an IMSI change in this case.

6.5.1 SIM-Initiated IMSI Change

This is the simpler of the two approaches. As discussed in Section 6.3, we suppose that the SIM has an application that decides when to trigger an IMSI change. When the logic encoded in the application decides to make a change, the SIM updates the EF_{IMSI} file with a new IMSI. The new IMSI will clearly need to be selected from the predefined list. How the list is used is a matter for the issuing network. For example, the IMSIs could be used in cyclic order or at random (or, more probably, pseudo-randomly). The SIM causes the ME to switch to the 'new' IMSI using the procedure described in Section 6.4.

6.5.2 Network-Initiated IMSI Change

In this case, the HN decides when to trigger an IMSI change. The HN will have a richer set of information to use to decide when to change IMSI than the SIM. For example, the HN could change the IMSI whenever the SIM changes SN or after a fixed number of calls.

As discussed in Section 6.3, when the HN decides to trigger an IMSI change, it must, by some means, send an instruction to the SIM. However, we have not been able to find a guaranteed secured channel in the current GSM specifications that could be used to transfer such an instruction to the SIM; as a result, this approach does not seem to work for GSM. However, this approach could be used in 3G and 4G mobile systems, as described in Chapter 9.

Note that modifications to GSM proposed in Chapter 5, allow the establishment of an authenticated channel between the HN and the SIM during authentication, since the modified GSM AKA ensures mutual authentication. We discuss this issue further in Chapter 9 (see Section 9.3.2).

6.6 Modifiable Multiple IMSIs

The second proposed means of deploying multiple IMSIs involves distributing new IMSI values from the HN to the SIM after its initial deployment, where the HN will choose each new IMSI from its pool of unused values. Such an approach clearly requires a means of communicating from the HN directly to the SIM. Like the scheme introduced in Section 6.5.2, this approach cannot easily be implemented in GSM because of the lack of a secure communications channel. However, we describe the use of such approach in 3G and 4G networks in Chapter 9.

In parallel work to that described here, van den Broek, Verdult and de Ruiter [163] present a similar multi-IMSI approach for GSM which does allow new IMSI values to be sent to the SIM after initial deployment. In their scheme, the subscriber's IMSI is replaced with a changing pseudonym called the *pseudo mobile subscriber identifier* (PMSI), which is only resolvable by the SIM's HN. The structure of a PMSI is the same as that of an IMSI, and it is treated like an IMSI by the SN which is unaware of the fact it is not an IMSI. They propose a scheme in which a new PMSI, a sequence number and a MAC are embedded together in the *RAND*, for transfer to the SIM (using the RAND hijacking technique introduced in Chapter 5). By embedding a MAC within *RAND*, they provide their own secure channel for communication from the HN to the SIM, addressing the issue discussed in the previous paragraph as a major obstacle to this approach. The form of their hijacked *RAND* has some similarities to the modified *RAND* employed in the scheme described in Chapter 5 (for very different purposes).

Interestingly, van Den Broek, Verdult and de Ruiter [163] also proposed modifications to 3G to achieve similar objectives; these are discussed in Chapter 9.

6.7 Experimental Validation

Testing the schemes is challenging due to the unavailability of a test network. However, the *SIMtrace* [169] hardware and software provided by the *Osmocom project*[1] enabled us to test the proposed modifications to the SIM. These tests are relevant both here and for the schemes described in Chapter 9; for reasons discussed below, tests by a USIM apply equally to a SIM. We used SIMtrace to monitor USIM-ME communications, together with *SysmoUSIM-SJS1*[2], a standards-compliant test UMTS UICC card, as our main test platform.

To validate the proposed modifications, we first ran an experiment to update the IMSI value in the test USIM. Using a standard contact smart card reader, smart card

[1] `http://osmocom.org`

[2] `http://www.sysmocom.de/products/sysmousim-sjs1-sim-usim`

scripting tool, and a custom script we were able to modify the IMSI value. Later, we developed a *SIM toolkit applet* using the Java card framework and the packages included with the 3GPP technical specification covering the USIM API for Java card [30]. The SIM toolkit applet used the *REFRESH* proactive command to cause the ME to fetch the new IMSI. We chose to use the *REFRESH* command as it is understood by all MEs which support proactive commands. To carry out the tests, we loaded the applet into the test USIM. We connected the test USIM and the ME to the SIMtrace device, which was connected to a laptop to record the APDUs exchanged between the test USIM and the ME. We tested the full range of modes of the *REFRESH* command, as discussed in Section 6.4. The observations from the experiment are as follows.

1. When a *REFRESH* command is executed in initialisation and file change notification mode, a series of read commands are issued by the ME. Although the record of APDUs exchanged showed that the IMSI file was read, we were unable to confirm that the read operation actually updated the IMSI value stored in the ME because we did not have access to a test network.

2. When the *REFRESH* mode is changed to SIM reset, the ME simply restarted its session, as expected. These observations confirmed that, if the *REFRESH* command is used in the SIM reset mode, the ME is made aware of the new IMSI. As a result, all future authentication procedures performed by the MS will use the new IMSI, which will have the effect of notifying the HN of use of the new IMSI.

During the experiments, we tested a range of standard MEs, all of which support the proactive command. As mentioned earlier, due to the unavailability of a test network we were unable to implement the modified HN. However, the changes required at the HN are purely software changes to the operation of the AuC database, which should not require significant additional computing resource.

Although we implemented our tests using a USIM, a SIM should give identical results, since both SIM and USIM understand the *REFRESH* proactive command and interpret it in the same way.

6.8 Analysis

The use of multiple IMSIs does not provide a complete solution to user identity confidentiality. While in use, the IMSI still functions as a pseudonym, potentially enabling the interactions of a single phone to be tracked for a period; of course, this is always

true for any mobile network when a subscriber resides in a single location area, even where only a privacy-preserving TMSI is used. Naturally, the more frequently IMSIs are changed the less the impact of possible tracking, but frequent IMSI changes have an overhead in terms of database management. The use of a predefined set of IMSIs further restricts the degree of user identity confidentiality protection. In this case, over a period of time it might be possible for an eavesdropper to link at least some of the fixed IMSIs. Overall the IMSI-changing proposal can be seen as allowing a trade-off between user privacy and the cost of implementing frequent IMSI changes.

Loss of IMSI synchronisation could be a critical issue. If, in the modifiable multiple IMSIs case, an active adversary is able to persuade the SIM to change its IMSI to an unauthorised value, then the SIM (and the MS) will cease to be able to access the network. It is therefore essential that robust cryptographic (and other) means are used to guarantee the correctness and timeliness of the new IMSI. In the predefined IMSIs schemes described in Section 6.5, loss of synchronisation cannot arise, as even if the SIM is persuaded to make an unauthorised change, the new IMSI will be known to the HN.

In the scheme described in Section 6.5.1, nothing changes in the authentication protocol, and so, the proposed scheme neither affects the existing security properties nor introduces any new concern. However, the scheme improves user identity privacy over the air interface.

6.9 Related Work

Apart from the recent work of van den Broek, Verdult and de Ruiter (described in Section 6.6), we know of no other proposed modifications to the operation of GSM with the objective of enhancing user privacy that do not involve major changes to the network infrastructure (as discussed in Section 4.6.2). However, a number of authors have proposed the use of multiple IMSIs for a single SIM, as summarised below.

Sung, Levine, and Liberatore [152] proposed a scheme to enhance location privacy which uses multiple IMSIs for a single SIM, and which has some similarities to the schemes we propose. However, their scheme involves an additional party in its operation, needs support by the ME, and requires wireless data connectivity for sending and receiving calls. The threat model is also very different, in that the HN is considered as a potential adversary. The scheme employs phones without a local SIM; instead the phone's software retrieves a virtual SIM offered by an Internet-accessible third party, which is used for a limited period and paid for using an anonymous Internet payment system in which messages are sent via Tor.

Tagg and Campbell [153] described a scheme to use multiple IMSIs for multiple networks with a single SIM. Their scheme involves the use of an update server to provide a suitable IMSI as and when it is required. The objective of their proposal is to avoid roaming charges by dynamically switching network provider. Marsden and Marshall [123] proposed a similar approach. The focus of their work is thus very different to the approaches introduced in this chapter; they also do not provide a means of transparently transferring new IMSIs to a SIM.

6.10 Summary

In this chapter we introduced two general approaches to using multiple IMSIs for a mobile subscriber. The goal of these approaches is to improve user privacy by reducing the impact of IMSI disclosure on the air interface. The proposed approaches provide a form of pseudonymity on the air interface, even when it is necessary to send the IMSI in cleartext, and hence reduce the impact of user privacy threats arising from IMSI capture.

We described a scheme for GSM which does not require any changes to the existing deployed network infrastructures, i.e. to the SN, air interface protocols or mobile devices. One major advantage is that the proposed scheme could be deployed immediately since it is completely transparent to the existing mobile telephony infrastructure. Further multi-IMSI privacy-enhancing schemes designed for use in 3G and 4G networks, are described in Chapter 9.

Part III

Privacy Issues in 3G and 4G

Overview

Part III addresses privacy threats arising in 3G and 4G. Novel schemes designed to address the threat of IMSI catchers in 3G and 4G are presented. Part III contains three chapters, as follows.

- *Chapter 7* reviews known privacy issues arising from disclosure of the permanent subscriber identity in 3G and 4G. It also briefly discusses previous attempts to address these issues.
- *Chapter 8* presents a critical analysis of the proposed modifications to the operation of 3G systems due to Arapinis et al. [48], intended to address threats to user identity privacy; it further proposes possible alternative means of mitigating these threats.
- *Chapter 9* describes and analyses novel schemes to address the decades-old privacy problem of disclosure of the permanent subscriber identity, focussing in particular on 3G and 4G systems.

Chapter 7

Privacy Issues in 3G and 4G

7.1 Introduction

The security features of 3G and 4G mobile systems represent a significant improvement on those of GSM, notably by providing mutual authentication between network and phone, and by incorporating integrity protection for signalling messages sent across the air interface. 4G systems, such as the *long-term evolution* (LTE) scheme, further enhance the data confidentiality feature across the air interface by separating the management of user data from that of signalling data, and by introducing a hierarchical key structure in which purpose-specific session keys are derived from K_{ASME} (the master session key, described in Section 3.4.1) to be used by the networking entities involved in cryptographic data protection. However, as discussed in Section 1.2, user identity confidentiality across the air interface has remained largely unchanged, relying in all cases on the use of temporary identities, and it has long been known that the existing measures do not provide complete protection for the user identity. In this chapter we describe the privacy properties of 3G and 4G, along with the known privacy threats arising from use of such systems. We further describe previous research directed towards addressing these threats.

The chapter is organised as follows. In Section 7.2 we briefly review key privacy terminology and the privacy features of the 3G and 4G air interfaces. Privacy threats arising from the use of 3G and 4G are briefly described in Section 7.3. Section 7.4 reviews previous attempts to address these privacy threats. Finally, in Section 7.5, we describe the motivation for the work presented in chapters 8 and 9.

7.2 User Privacy

7.2.1 Privacy Terminology

We start our consideration of user privacy by introducing some privacy-related terminology. Unless otherwise stated we follow RFC 6973 [72].

- *Anonymity*: The property for a user that he or she is not identifiable by an observer within a set of users, known as the *anonymity set*.
- *Unlinkability*: Two or more items of interest (e.g. users, messages, actions) are unlinkable if an interested party cannot distinguish whether or not they are related.
- *Untraceability*: The property for an object that a third party cannot determine whether or not the object exists or is present. In other words, an observer cannot follow an object as it moves from one location to another [61].
- *Pseudonymity*: The property for an individual that he or she is identified by a pseudonym. Pseudonymity is enhanced, i.e. the pseudonyms are less likely to be linkable, if:
  - less personal data can be linked to the pseudonym,
  - the same pseudonym is used less often and across fewer contexts, and
  - independently chosen pseudonyms are more frequently used for new actions.
- *Identity confidentiality*: The property for an individual that only an authorised party can identify the individual within a set of other individuals.

User anonymity, user unlinkability and user untraceability are closely related, and are clearly desirable from a user privacy perspective.

7.2.2 Privacy Features

The privacy properties of 3G and 4G mobile systems are listed below. Further details can be found in 3GPP TS 33.102 [31].

- *User identity confidentiality*: The property that the permanent user identity (IMSI) of a user to whom service is being delivered cannot be learnt by monitoring the radio access link; this privacy feature ensures user *anonymity* for the air interface. This is the same as the IMSI confidentiality feature of GSM, described

in Section 4.2.1. The mechanism used to protect the confidentiality of the user identity in 3G and 4G is the same as that used in GSM. The only difference is that 4G extends the length of the temporary user identity to accommodate more system information. Although this change improves flexibility in network operations, it does not prevent IMSI catching attacks. The scenarios in which the IMSI is disclosed on the air interface, described in Section 4.2.1, also apply to both 3G and 4G, and so, with respect to the robustness of *user identity confidentiality*, 3G and 4G are the same as GSM.

- *User location confidentiality*: The property that the presence or arrival of a user in a certain area cannot be determined by eavesdropping on the radio access link. Since location information in 3G and 4G is easily accessible, the robustness of this property largely depends on how well 3G and 4G protect the confidentiality of the user identity.

- *User untraceability*: 3GPP [31] defines this to be the property that an intruder cannot deduce whether multiple services are delivered to the same user by eavesdropping on the radio access link; by definition, this privacy feature also guarantees *user unlinkability*. Like *user location confidentiality*, to ensure *user untraceability* 3G and 4G should protect the confidentiality of the user identity.

To provide the above privacy properties, both 3G and 4G use a pseudonym as a temporary user identity in place of the IMSI when communicating across the air interface. To avoid user tracking, the temporary user identity is refreshed after a certain period of time, configurable by the SN. In addition it is required that any signalling or user data that might reveal the user's identity must be sent via an encrypted channel on the radio access link.

7.3 Privacy Threats

In this section we continue the discussion of privacy threats introduced in Section 4.5. We focus on those privacy threats relevant to 3G and 4G.

7.3.1 IMSI Catching Attack

The IMSI catcher, described in Section 4.5.3, is a persistent threat to all generations of mobile systems. We defined IMSI catching attacks in Section 4.5.3. Although IMSI catchers have mostly been discussed in the context of GSM, their presence in 3G and 4G networks is also feasible. It seems likely that, in practice, active attackers with

a fake base station capability force 3G- and 4G-capable mobile devices to use GSM protocols in order to maximise the attack potential.

An IMSI catcher will typically use the configuration data for a genuine network to set up its fake base station. The fact that the broadcast system information messages used in 3G and 4G are not cryptographically protected allows an active attacker to emulate a 3G or a 4G mobile network; such an attacker can request a mobile device to transmit its IMSI. Since requesting an IMSI is a legitimate network activity (see Section 4.2.1), the attacker can easily obtain the IMSI of a target user. The feasibility of IMSI catchers has been practically demonstrated [63, 147], and widely discussed [100]. IMSI catchers for 3G and 4G are commercially available[1].

3GPP technical documents TR 33.821 ([14], Section 5.1.1) and TS 21.133 ([3], annex A.1) discuss the IMSI catcher threat. However, completely removing the threat of a fake base station is very difficult; communications between a base station and an unidentified ME are almost inevitably unprotected, since no key is available on which to base cryptographic protection. The use of asymmetric cryptography could help significantly, but the introduction of such technology would involve major changes to network architectures and would also significantly increase implementation complexity. However, advances in hardware technology and the availability of software [167, 171] have made IMSI catcher capabilities widely available, which may force regulatory bodies to take steps to oblige network operators to reduce the threat [132, 146]. Technical details of the operation of IMSI catchers have been widely discussed in the academic literature [65, 135, 151].

7.3.2 IMSI Paging Attack

This attack exploits the same flaw that enables the IMSI paging attack in GSM (see Section 4.5.5). Although 3G and 4G provide integrity protection for many signalling messages, *paging type 1* messages, which contain the subscriber's IMSI, are not integrity protected (see Section 3.6.3). Thus, an active adversary can introduce spurious paging messages into the network to both detect the presence of a UE with a specific IMSI, and also learn the current TMSI for this device, analogously to the IMSI paging attack in GSM.

7.3.3 User Linkability Attack

This threat was uncovered by Arapinis et al. [48, 49], and is much like the GSM user linkability attack described in Section 4.5.4. It exploits the error messages associated

[1] `http://www.pki-electronic.com/products/interception-and-monitoring-systems/`
`http://rayzone.com/en.piranha.html`

with 3GPP AKA, as described in Section 3.4.2. When a USIM runs AKA, it first verifies the MAC value, and if MAC verification is successful it performs a further check on the *SQN* value (see Section 3.4.1). If the MAC verification fails an error message is returned to the base station and processing stops; if the *SQN* verification fails then a different error message is sent. The fact that the two failures give rise to two different error types allows an active adversary to gain user-linkable information from the USIM's response to a replayed authentication request.

Suppose an active attacker has intercepted a genuine (*RAND*, *AUTN*) pair sent to a target UE, and is interested in tracing this UE. If this pair is sent by a fake base station to a specific UE as part of the AKA protocol, there are two possible outcomes (see Figure 7.1). If the recipient UE is the device to which the (*RAND*, *AUTN*) pair was originally sent, then it will respond with an authentication failure containing a *sync-failure* error code, indicating a *SQN* check failure. This reveals the presence of the target UE. Otherwise, if the recipient UE is not the target UE, it will respond with an authentication failure containing an error code indicating *MAC-failure*. That is, the error code can be used to distinguish between a target UE and other UEs.

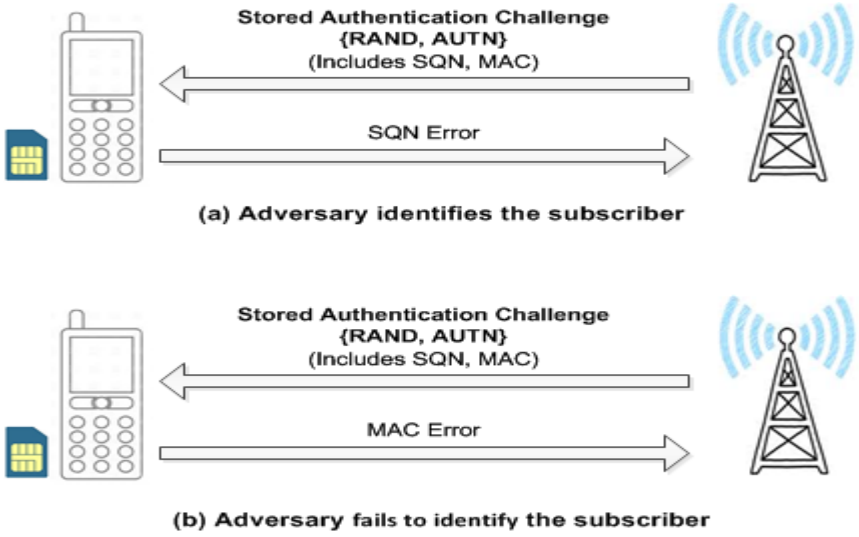

Figure 7.1: Possible outcomes in a user linkability attack

The attack could be used to profile user movement within a restricted area, such as in an office building. Given more resources, the attack could be extended to trace the movements of a target UE in a larger area, e.g. within a city.

7.4 Addressing the Threats

We now review the extensive prior art aimed at improving user privacy in mobile networks, focussing in particular on user identity confidentiality in 3G and 4G networks. As in Section 7.2.2, *user identity confidentiality* is provided by using a temporary subscriber identity when communicating across the air interface, as soon as it is available. Although this approach provides user identity confidentiality against a passive adversary, it has serious vulnerabilities in the presence of an active adversary. Over the last two decades this weakness has motivated many proposed modifications to existing network protocols aimed at providing more robust protection for the confidentiality of the IMSI. We described some of this work in Section 4.6.2, focussing primarily on GSM-specific proposals. In this section we describe a range of research addressing this privacy problem in the context of 3G and 4G networks.

Before reviewing the academic research, we note that 3GPP technical report TR 33.821 ([14], Section 5.1.1.2) outlines two general approaches to addressing the identity confidentiality vulnerability arising from the need to send the IMSI across the air interface. The first is to use another type of pseudonym to replace the IMSI, where this pseudonym is managed between the USIM and the HN. The second is to use public key cryptography, i.e. to enable the mobile device to encrypt the IMSI using a public key belonging to the SN prior to transmission across the air interface. A range of more detailed proposals for both approached have been made, and are discussed in Sections 7.4.1 and 7.4.2 below. We also consider another general approach to addressing the issue, namely IMSI catcher detection, in Section 7.4.3.

7.4.1 Asymmetric Cryptography Based Schemes

The main reason the IMSI needs to be sent across an unencrypted air interface link is that, until the mobile device and SN have established a shared secret key using AKA, there is no key available to perform encryption; thus asymmetric encryption of the IMSI using a network public key seems the obvious solution to IMSI disclosure. Over the last two decades a range of schemes using asymmetric key cryptography have been proposed to try to address the IMSI disclosure threat. We summarise below some of the proposals of this type.

- Many authors (including [48, 49, 51, 76, 92, 94, 175, 177] and Section 9.4 of [127]) have proposed modifications to AKA and UE identification using public key cryptography. However, all such modifications require the integration of *public key infrastructures* (PKIs) into mobile networks, necessitating significant changes to the operation of the HN, SN, and UE.

- Køien [113] proposed the use of identity-based encryption to encrypt the IMSI for transmission across the air interface. Whilst the use of identity-based cryptography avoids the need for a PKI, the scheme nevertheless involves significant changes to all the system components. A UE has to compute a public identity for each SN it visits, and an SN has to record the private identity, computed by the respective HN, for each HN it interacts. As a result, it appears that the scheme is not deployable with existing systems.

Thus, all the previously proposed privacy-enhancing system enhancements employing asymmetric key cryptography require significant changes to all the main components of the system, including HNs, SNs, phones and USIMs. As we have observed previously, making such modifications to the existing mobile system infrastructure is almost certainly impractical in practice; thus the only possible scenario in which such schemes might find a use is in future generation mobile networks.

7.4.2 Pseudonym-Based Schemes

Over the last 20 years, many authors have proposed pseudonym-based enhancements to 3G and 4G network protocols to try to address the air interface IMSI disclosure problem. We summarise below some of the key proposals of this type.

- In the mid-1990s, Mitchell [128] proposed a pseudonym-based user identity protection scheme for 3G to overcome the shortcoming of disclosure of the permanent user identity during the initial registration of a mobile device with an SN. The scheme introduces $TMUI_S$ — a temporary user identity managed by the subscriber's HN. $TMUI_S$ is used instead of the IMSI to identify a user to the HN. A UE uses $TMUI_S$, along with the TMSI[2], to identify itself across the air interface. When the UE has a valid TMSI, it identifies itself using this TMSI; otherwise, it uses the $TMUI_S$ as its identity. On receiving the $TMUI_S$, the SN forwards it to the HN, and the HN generates a new $TMUI_S$, masks it, and sends to the SN along with the authentication data (analogous to an AV). The SN forwards the received $TMUI_S$ to the UE as part of subscriber authentication. The scheme was proposed before the deployment of 3G, and so it incorporates the use of an authentication protocol slightly different to UMTS AKA. Maintaining synchronisation of the $TMUI_S$ between the UE and the HN is not described, and analysis of the efficiency of the scheme is kept as future work. Like most of the many subsequently proposed schemes of this type, implementing such a pseudonym system

[2]Although Mitchell uses the term $TMUI_N$, the role of $TMUI_N$ is the same as the TMSI in 3G.

in 3G would require significant changes to the air interface protocol, and would affect the operation of all the 3G entities. For further details, see also Section 9.3 of [127].

- Barbeau and Robert [51] present two different pseudonym-based schemes relying on the use of a one-time alias that is much like an IMSI. Their first scheme introduces an HN-managed *coupon*[3] to be used in authentication instead of both the IMSI and the TMSI. Each coupon corresponds to an IMSI known to the HN. The HN transfers a new coupon via an encrypted channel to the UE in each authentication session, and the UE uses this coupon in the next service request. The researchers acknowledge a major drawback of the scheme, namely that it modifies the message format in the authentication protocol, requiring changes to the SN, HN and UE.

  The second scheme Barbeau and Robert propose uses a one-time alias, the *international mobile anonymous number* (IMAN), that is derived independently by both the HN and the UE; this allows the scheme to avoid any changes to the operation of the intermediate network entities. The IMAN is used instead of both the IMSI and the TMSI by the SN. The HN keeps record of the current and old IMAN corresponding to an IMSI. The IMAN is computed as $IMAN = MD5(AK \parallel SQN \parallel RAND)$, where MD5 is the well known cryptographic hash function [124], AK is the anonymity key, SQN is the sequence number, and $RAND$ is the random number (all as in 3G AKA). The output of the MD5 function is truncated to the length of an IMSI. When the HN generates AVs, it may need to repeat the IMAN computing procedure several times with different $RAND$ values to find an IMAN that is not already in use. The IMAN is updated by both the UE and the HN in every successful authentication, although it is not clear how the HN will be aware of which specific AV was used in the immediate past successful user authentication. This information is required to enable the HN to update the IMAN correctly. Moreover, how the IMAN will be transferred from the USIM to the ME to be used by the UE is not discussed in the scheme. The researchers address the synchronisation of the IMAN between the HN and the UE, and claim self-synchronisation of the HN and the UE. However, they do not consider scenarios in which IMAN synchronisation fails, for example, if an AV is lost or not used the HN will update the IMAN but the UE will not. Thus, although this scheme has the major advantage of only requiring changes to the USIM and the HN, it cannot manage possible identity desynchronisation,

[3] The structure of the coupon is unspecified

which is critical to the operation of a 3G network.

- Sattarzadeh, Asadpour and Jalili [144] introduce an improved user identity confidentiality mechanism for 3G that replaces use of the IMSI with *anonymous tickets*. The HN stores two tickets for each subscriber, and manages the relationship between the tickets and the IMSI. The scheme changes the operation of 3G AKA to incorporate new messages, which requires changes to the operation of the UE, SN and HN.

- Juang and Wu [108] propose a modified 3GPP authentication protocol intended to ensure user identity confidentiality. Their scheme is based on the work of Zhang and Fang [176], an enhancement to 3G AKA addressing other issues. When a UE is required to send the IMSI to the network, the UE masks the IMSI with a *token*, appends the random number used to compute the *token* to the masked IMSI, and sends the combination of the masked IMSI and the random number to the HN via the SN. On receiving this value, the HN computes the *token* as $H(x \parallel r)$, where x is a new master key only known to the HN, r is the received random number, and H is a MAC function. The HN checks the validity of the request, and uses the computed *token* to retrieve the IMSI of the subscriber. The HN computes a new *token* using a new random number, and sends the *token* in encrypted form with the random number in cleartext to the SN, and the SN forwards these values to the UE in the next authentication request. After successful authentication the UE updates its stored *token* and the random number. Since the UE appends the random number used to compute the *token* to the masked IMSI and the random number is transmitted in cleartext, the relation between the old and the new random number could be learnt by a passive adversary; hence, this random number could be used to track a user. The scheme modifies the AKA messages, and hence involves changes to the operation of the SN, HN and UE.

- Choudhury, Choudhury and Saikia [71] proposed a scheme using a frequently changing *dynamic mobile subscriber identity* (DMSI) instead of the IMSI across the air interface. The DMSI is constructed by concatenating the MCC, the MNC, a random number chosen by the HN, and a 128-bit encrypted version of the chosen random number. As a result, the structure and length of the DMSI differs significantly from the IMSI. The variable part of the DMSI is transferred to the UE during authentication, and is updated on every run of the authentication protocol. The HN keeps a record of the set of DMSIs allocated to each user, and the USIM updates its DMSI on receiving a new value after completing authentication. The use of the DMSI requires changes to the SN, HN and UE.

Thus, almost all the previous proposals for pseudonym-based schemes require changes to all the main components of the system, including HNs, SNs, phones and USIMs. This essentially means that deploying any of these solutions would require deploying an entirely new network infrastructure, which seems unlikely to occur. The only scheme which does not require such a major redeployment is the second scheme of Barbeau and Robert [51]; however, as we have discussed, this scheme appears to permit loss of synchronisation between the USIM and the HN, which is likely to lead to a permanent loss of service. As a result, this solution too is unlikely to be deployable in practice.

7.4.3 IMSI Catcher Detection

The schemes described above try to avoid any threats (active or passive) arising from disclosure of the IMSI on the air interface. Another approach to addressing this threat is to detect the presence of an IMSI catcher. Tools to achieve this are known as *IMSI-catcher-catchers*.

Debrowski et al. [73] present two independent implementations of an IMSI-catcher-catcher, i.e. a hardware-based stationary device and a mobile application, intended to detect the presence of an IMSI catcher. Other such mobile applications include *Snoopsnitch* [133, 170], *Darshak* [64, 165], and *Android IMSI catcher detection*(AIMSICD) [164]. These applications have major limitations, i.e. they are supported only on the android platform and two of them are hardware-dependent. In general these applications continuously analyse available network information, scan for any anomalies in the network, and alert the user when anomalies are detected. However, many of the anomalies detected by these applications could be part of normal network operation; hence, the alerts could be false positives. Moreover, these applications simply warn the user of abnormal activities, and do not prevent a mobile device connecting to a potential IMSI catcher.

7.5 Research Motivation

Although 3G and 4G provide the identity confidentiality feature using temporary subscriber identities, as we discussed in Section 7.3.1, this mechanism does not work in the presence of an active adversary such as an IMSI catcher. The IMSI is also sometimes sent for entirely legitimate reasons, meaning that the identity confidentiality mechanism is vulnerable even to a passive interceptor; however, the threat from IMSI catchers is significantly greater since it allows the IMSI to be revealed at will. It is therefore vital that any scheme designed to protect the subscriber's IMSI should be effective

against both passive and active adversaries; also, for practical reasons, any enhancement to existing mobile systems designed to enhance this feature needs to work with the existing system infrastructure.

As discussed in Section 7.4, over the years a wide range of possible solutions to the IMSI catcher problem have been proposed, which vary in their use of cryptographic techniques. Introducing new cryptographic techniques could solve the problem of the IMSI catcher; however, such schemes requires significant changes to the deployed infrastructure, the USIM, and the phone. It seems likely that making such modifications to widely deployed mobile systems is almost certainly impractical in practice. In addition, full implementation details of proposed schemes have not been provided. These observations motivated a critical examination of a recently proposed set of modifications to 3G. This is the focus of the next chapter.

Since pseudonym-based schemes typically use only the cryptographic techniques already in use, such an approach is a candidate for use in enhancing existing mobile systems; unfortunately, as discussed in Section 7.4.2, almost all such schemes require significant modifications to the air interface protocol, which would require changes to the operation of all the SNs as well as all the deployed phones. As a result, it seems likely that making the necessary major modifications to the operation of the air interface after deployment is infeasible in practice. It would therefore be extremely valuable if a scheme to reduce the threat from IMSI catchers and thereby better protect user privacy could be devised, which does not require significant changes to the existing network infrastructures and has minimal computational cost. This observation motivates our efforts to devise novel schemes to improve user identity privacy in mobile systems which could actually be deployed. We describe the novel schemes in Chapter 9.

Chapter 8

Another Look at Privacy Threats in 3G

8.1 Introduction

The 3GPP standards, which incorporate a range of security features [3, 31], are the basis for a large part of the world's mobile systems. As a result, any security or privacy flaws identified in these standards potentially have major implications. In this chapter we are primarily concerned with one particular feature of 3G security, namely the service known as *user identity confidentiality* (see Section 7.2.2). This service seeks to minimise the exposure of the IMSI on the air interface. The main security feature incorporated into the 3G system designed to provide this service is the use of frequently changing temporary identities, which act as pseudonyms.

A recently published paper by Arapinis et al. [48] describes two novel attacks on this service, which enable user device anonymity to be compromised. As well as describing the two attacks, modifications ('fixes') to the protocol are described which aim to prevent the attacks, and verifications of these fixes using ProVerif are also outlined.

In this chapter we re-examine these proposed fixes, and briefly describe the findings. We find significant shortcomings in the proposed fixes, and suggest possible alternative approaches to some of the modifications. We argue that some of the weaknesses in user identity confidentiality are impossible to fix, meaning that making significant system changes to address some of them are unlikely to be worth the effort. We also demonstrate certain limitations in the effectiveness of tools such as ProVerif if not used with appropriate care, and in particular if they are used without a detailed understanding of the cryptographic primitives being employed. The discussions here also apply to 4G, although we use 3G terminology throughout. The main content of this

chapter was presented at ACISP 2014, and has been published in the proceedings [109].

The remainder of the chapter is structured as follows. In Section 8.2 the attacks of Arapinis et al. are summarised, together with a description of their proposed fixes. Sections 8.3 and 8.4 provide analyses of these fixes. Finally, the findings are summarised and conclusions are drawn in Section 8.5.

8.2 Privacy Threats and Fixes

8.2.1 The Attacks

Arapinis et al. [48] describe two apparently novel attacks that breach user identity confidentiality in 3G mobile systems. These two threats operate as follows.

- *IMSI paging attack*: This attack exploits a *paging* message (or, more formally, a *paging type 1* message — see Section 3.6.3). Such messages are sent from the network to all mobile devices in a particular area to establish a radio connection to a specific UE, and can contain either an IMSI or a TMSI. Most importantly, paging messages are not integrity protected (see Section 3.6.3), and hence an active adversary can introduce spurious paging messages into the network to both detect the presence of a UE with a specific IMSI, and also to learn the current TMSI for this device (see Section 7.3.2). This poses a threat to mobile identity privacy.

- *User linkability attack*: This attack exploits the error messages incorporated into the AKA protocol, as described in Section 3.4.2. Suppose an attacker has intercepted a genuine ($RAND$, $AUTN$) pair sent to a particular UE. As discussed in Section 7.3.3, if these values are replayed to a specific UE at some later time, two possible error responses will arise depending on whether the intercepted ($RAND$, $AUTN$) pair was intended for this UE. That is, the error code can be used to determine whether or not a UE is the same as a target UE, and this is clearly another means of breaching user identity confidentiality.

8.2.2 Observations

We start by observing that the first threat, whilst apparently novel, is closely related to another threat to user identity privacy. As described in Section 6.2 of 3GPP TS 33.102 [31], 'when the user registers for the first time in a serving network, or when the serving network cannot retrieve the IMSI from the TMSI by which the user identifies itself on the radio path', the serving network must obtain the IMSI from the UE — this

is performed using a *user identity request/user identity response* message pair, where the latter message contains the IMSI. 'This represents a breach in the provision of user identity confidentiality' [31]. This attack, called *user identity catching* or *IMSI catching* (see Section 7.3.1), is further mentioned in A.1 of 3GPP TS 21.133 [3], and is also noted by Arapinis et al. ([48], Section 2.2).

Given that this attack has long been known, i.e. an active attacker can obtain the IMSI of any UE by impersonating the network, neither of the new attacks appear to significantly further weaken the user privacy service. That is, neither of the new attacks appear to be any easier to launch than the IMSI catching attack — in particular, they both require active impersonation of the network as is the case for an IMSI catching attack.

Most interestingly, the second attack seems to be an issue that has not previously been discussed in the literature. It is just one example of a very broad class of threats arising from poorly designed error messages that reveal information of value to an attacker — see, for example, Vaudenay [172].

8.2.3 The Fixes

As well as describing the two privacy issues, Arapinis et al. [48] give three separate modifications to the operation of 3G systems designed to fix the two newly identified problems as well as the well known user identity catching attack. We next briefly describe these proposed modifications.

- *Fixing the IMSI paging attack*: This modification is not described in complete detail ([48], Section 5.2), and as a result some suppositions need to be made. It involves cryptographically protecting the paging message using a secret key UK known only to the network and the UE. Like CK and IK, this additional key is generated as a function of the $RAND$ and K by the HN's AuC, and is provided to the SN as part of an extended AV.

  The paging message format is modified to incorporate two additional fields, namely a sequence number SQN and a random challenge $CHALL$. It is not clear whether SQN is in the same 'series' as the SQN sent in the $AUTN$, or whether this is a distinct sequence number used for this purpose only. This issue is discussed further in Section 8.3 below.

  The entire paging message is then encrypted using UK. However, the method of encryption is not specified. This issue is also discussed further in Section 8.3 below.

Since this message is broadcast, it is received by all UEs currently attached to a base station. Each UE must use its current *UK* to decrypt the message. By some (unspecified) means the recipient UE decides whether or not the decrypted message is intended for it — Arapinis et al. simply state ([48], Section 5.2) that each UE 'has to decrypt and check all the received IMSI paging to determine if it is the recipient' (sic). If it is the intended recipient, then the UE checks the *SQN* against its stored value to verify its freshness (as in AKA). If it is fresh then the USIM updates its stored *SQN*, and sends a paging response containing the TMSI and the received value of *CHALL*; otherwise, if the freshness check fails, the paging message is ignored.

- *Fixing user linkability attack*: This fix involves leaving the 'normal' operation of AKA unchanged; the only modification is to require (asymmetric) encryption of authentication failure report messages, thereby hiding the nature of the embedded error message. This encryption is performed using a public encryption key belonging to the SN. Providing a reliable copy of this key to the UE requires the pre-establishment of a *public key infrastructure* (PKI) involving all the 3G network operators, in which each network operator has an asymmetric encryption key pair *and* a signature key pair. Each operator must use its private signature key to create a certificate for every other network's public encryption key. Every USIM must be equipped with the public signature verification key of the issuing (home) network.

  In order for the UE to obtain a trusted copy of the appropriate public encryption key, the SN must send the UE a copy of a certificate for its public encryption key, signed using the private signature key of the USIM's home network (this could be achieved by modifying an existing signalling message or by introducing a new such message). The USIM exports its trusted copy of the public verification key of its home network to the phone, and the phone can use this to verify the certificate, thereby obtaining the required trusted public encryption key. The phone can perform the encryption of the failure report message, obviating the need for the USIM to perform any computationally complex asymmetric encryption operations.

  A further modification to the failure report message is proposed by Arapinis et al. [48], namely to include the USIM's current value of *SQN*. This change is designed to enable resynchronisation of this value by the network, but is not explained further.

- *Fixing user identity catching*: Finally, Arapinis et al. [48] also propose modify-

ing the procedure by which a UE identifies itself when first joining a network. They propose that the UE asymmetrically encrypts the *user identity response* message containing the IMSI. As in the previous modification, this encryption is performed using the public encryption key of the SN.

8.3 IMSI Paging Re-Examined

There are a number of significant issues with the fix proposed to mitigate IMSI paging attacks. We enumerate some of the most serious.

1. Introducing a new type of session key, i.e. the UK, has major ramifications. Since the SN does not have access to the long term key K, the session key UK will need to be generated by the HN's AuC and sent to the SN as part of the AV. Introducing an additional key type means that the authentication vectors will need to become 6-tuples to include the UK value, which will involve changing the formats of messages sent between networks (this is, in itself, a significant change).

2. As noted in Section 8.2.3 above, there are two possible ways in which the SQN might be generated and managed. It could be generated and verified using the same mechanism as employed for the AKA protocol, or a separate sequence number scheme could be involved. Unfortunately, there are major implementation difficulties with both options.

   (a) Using the same SQN values as are used in the AKA protocol is problematic. The SN does not have a means of finding out these values, as they are not included in the authentication vectors sent to the SN. Even if the current SQN value was sent as part of the authentication vector (which would affect the inter-network signalling infrastructure), two major problems remain. Firstly, if the SN is permitted to generate new SQN values and have them accepted by the USIM, then this means that the SN is able to modify the SQN value stored by the USIM. This could have the effect of invalidating any unused authentication vectors that the SN retains for the UE. Secondly, giving the SN the power to change the SQN value held by the USIM is a major change in the current trust model, and would give the SN the power to, deliberately or accidentally, completely block the operation of the USIM by sending it a very large SQN value.

   (b) Using a different SQN value also raises major issues, as there is no obvious mechanism to keep multiple networks aware of the current value of the

SQN for a particular UE. This would require the HN to maintain the current value, and for serving networks to exchange messages with the HN to maintain synchronisation between the value held by the USIM and the HN.

3. The 'encryption' of the paging message appears to be intended to provide two distinct security services:

   (a) guarantees to the recipient regarding the origin and integrity of the message, and

   (b) confidentiality of the contents so that passive interceptors cannot observe the link between an IMSI and a TMSI.

   It is well known that simple encryption cannot guarantee property (a), especially if that means use of a stream cipher (see, for example, Section 9.6.5 of Menezes, van Oorschot and Vanstone [124]). However, stream cipher encryption is the only encryption primitive available in the current 3G security architecture. Clearly what is really required is the application of an authenticated encryption technique [104], which would provide the necessary security guarantees. However, this is never made explicit by Arapinis et al. [48]. Their success in proving the security of the modification using ProVerif suggests that their input to ProVerif implicitly assumed the provision of properties (a) and (b), whereas their description of the necessary modifications to the system did not make these requirements explicit. This shows the danger of not carefully considering and making explicit all the properties of the cryptographic primitives being employed.

   Of course, the SN and UE share a pair of keys (*CK* and *IK*) designed explicitly for confidentiality and integrity protection of data and signalling messages. A much simpler solution, which achieves precisely the same objectives, would be to first encrypt the paging message using *CK* and then generate an accompanying MAC using *IK*. This would both achieve the security objectives and avoid the need to introduce an additional key type.

4. Finally, we note that, even if it could somehow be repaired, the fix imposes very significant burdens on the system. As stated by the authors (final sentence of 5.2 of [48]) the overheads of the proposed modification are non-trivial. This is because every UE that receives a paging message is required to decrypt it and somehow verify whether or not it is intended for them.

In conclusion, the number and seriousness of the issues identified with the fix, especially relating to the use of the sequence number *SQN*, suggest that it cannot work

in practice. Moreover, finding an alternative fix without completely redesigning the 3G system appears highly problematic. As a result it would appear that accepting that user identity confidentiality is imperfect seems inevitable, a point we return to below.

8.4 User Linkability and Identity Catching Re-Examined

In evaluating the fix proposed to address the user linkability attack, we start by considering the practicality of introducing a brand new PKI. Whilst the required PKI is relatively small scale, involving only the network operators, introducing such a PKI would nevertheless involve significant changes to the operation of the system. In particular, over and above requiring changes to all phones, all USIMs and all networks, every USIM would need to be equipped with a public key, every network would need to exchange public keys and certificates with every other network, certificates (potentially quite large) would need to be routinely sent across the air interface, and the USIM would need to routinely transfer a public key to its host phone (across a smart card interface with a very limited data transfer capability). That is, whilst the PKI itself might be relatively small-scale, the changes to the air interface protocol to allow its use would require fundamental changes to the system infrastructure. It is not even clear how a phased deployment could be undertaken, and changing the entire system (including all mobile phones) at a single point in time is clearly infeasible.

It is interesting to note that the difficulty of providing robust identity privacy without asymmetric cryptography has long been known — see, for example, Mitchell ([130], Section 4.1). Indeed, this point is also made by Arapinis et al. ([48], Section 5.5) who make similar remarks. This suggests that modifications analogous to the proposed fix have been considered in the past, and rejected for reasons of complexity and low pay off (a point previously discussed in Section 7.4.1).

Moreover, deploying the required PKI requires all networks to possess two key pairs, one for encryption/decryption and one for signature generation and verification. This is because, in general, the widely accepted principle of *key separation* (see, for example, 13.5.1 of Menezes, van Oorschot and Vanstone [124]) requires that different keys are used for different purposes. However, if sufficient care is taken, sometimes the same key pair can be securely used for both encryption and signature, although this is not without risks (see, for example, Degabriele et al. [74]).

We further note that if the private decryption key of any network is ever compromised, then security is compromised. The usual solution in a PKI is to deploy a revocation system, e.g. in the form of *certificate revocation lists* (CRLs). However, deploying CRLs on the scale necessary would appear to be very challenging in a 3G mobile

system. Indeed, the difficulties of deploying CRLs across networks are well-established, [112, 131].

One alternative to the proposed solution would simply be to remove the error code from the error message, or, to minimise protocol modifications, to program mobile phones to always return the same error message regardless of how AKA actually fails. This is, in any case, clearly best practice for any security protocol, i.e. if an authentication procedure fails then the only information that should be provided is that the process has failed, and not how.

Finally we note that implementing the proposed fix to mitigate IMSI catching is problematic. Requiring a UE to encrypt the IMSI it sends to the network requires the phone to have a reliable copy of the network's public key. This will, in turn, require the network to send the UE a certificate — but which one? The UE will only be able to verify a certificate signed by the USIM's HN, but the SN will not know what this is until it has seen the IMSI. That is, the UE will not be able to encrypt the IMSI for transmission to the network until the network knows the IMSI, and hence we have a classic 'chicken and egg' problem.

8.5 Summary and Conclusions

It would appear that the modifications proposed to address the identified privacy threats either do not work or impose a very major overhead on the network, over and above the huge cost in modifying all the network infrastructure. Very interestingly, the failures in the fixes arise despite a detailed analysis using formal techniques.

Of course, as discussed in the previous chapter, making significant changes to a protocol as widely deployed as the 3G air interface protocol is unlikely to be feasible, so the discussion here is perhaps rather moot. However, even where the fixes appear to work, in two cases significantly simpler approaches appear to have been ignored. That is, removing the error messages would mitigate the user linkability attack (and would also conform to good practice), and it would appear that the introduction of a new key UK is unnecessary. If changes are to be made, then it is vital to try to minimise their impact on the operations of the system.

Most significantly in any discussion of whether it might be worth trying to implement 'fixed up' versions of the proposed modifications, there exist 'passive' attacks on user identity confidentiality other than those discussed thus far. For example, a malicious party wishing to discover whether or not a particular phone is present in a cell could simply inaugurate a call to the phone or send it an SMS, simultaneously monitoring messages sent across the network (an example of a semi-passive attacker,

as introduced in Section 4.3). If such a procedure is repeated a few times, then it seems likely to be sufficient to reveal with high probability whether a particular phone is present, especially if the network is relatively 'quiet'. Such an attack only requires passive observation of the network, and hence would be simpler to launch than attacks requiring a false base station (which is the case for all the attacks we have discussed previously). Moreover, addressing such an attack would be almost impossible.

We can thus conclude that not only are the proposed fixes highly problematic, but providing a robust form of user identity confidentiality is essentially impossible in practice. That is, if highly robust identity confidentiality is not achievable, then it is unlikely to be worth the huge cost of making changes of the type proposed. The 'pay off' in mitigating some threats but not others is small relative to the overall cost of implementing them. Nevertheless, it may still be worth mitigating the most serious security and privacy threats to currently deployed mobile systems if it can be done in a cost-effective way, an observation that motivates the work described in the remainder of this thesis.

Finally, the practical and security issues encountered in considering the detailed implementation of the proposed modifications suggests that the use of formal tools to try to guarantee security and privacy properties should be performed with great care. In particular, any such analysis should always be accompanied by an analysis of the practical working environment for the security protocol.

Chapter 9

Trashing IMSI Catchers

9.1 Introduction

Although a number of possible modifications to 3G and 4G protocols to enhance user privacy have been proposed, as described in Section 7.4, unfortunately, they all require significant alterations to the existing deployed infrastructures, something that is almost certainly impractical to achieve in practice. In this chapter we describe novel authentication schemes for 3G and 4G systems designed to defeat IMSI catchers in a way that does not impose major infrastructure changes. Our first scheme makes use of multiple IMSIs for an individual USIM to offer a degree of pseudonymity for a user. The second scheme prevents disclosure of the subscriber's IMSI by using a dynamic pseudo-IMSI that is only identifiable by the subscriber's home network. A major challenge in using pseudonymous IMSIs is possible loss of identity synchronisation between a USIM and its home network, an issue that has not been adequately addressed in earlier work. We present an approach for identity recovery to be used in the event of pseudo-IMSI desynchronisation.

Both schemes require changes to the home network and the USIM, both owned by a single entity in the mobile systems, but not to the serving network, mobile phone or other internal network protocols, enabling simple, transparent and evolutionary migration. We provide analyses of the schemes, and verify their correctness and security properties using ProVerif. The first scheme described in this chapter was presented at SSR 2015, and has been published in the proceedings [110]. It was also presented at an internal conference of the *Fraud and security group* (FASG)[1] of the *GSM association* (GSMA).

The remainder of the chapter is structured as follows. The threat model for the

[1] `http://www.gsma.com/aboutus/leadership/committees-and-groups/working-groups/fraud-security-group`

schemes is described in Section 9.2. Sections 9.3 and 9.4 provide descriptions of two approaches to the use and management of multiple IMSIs in a USIM. This is followed in Section 9.5 by an analysis of the second approach, namely the modifiable multiple IMSIs scheme. In Section 9.6, a version of the pseudonymous IMSI scheme robust against loss of identity synchronisation is described. An analysis of the robust pseudo-IMSIs scheme is provided in Section 9.7. In Section 9.8, we present a ProVerif-based formal verification of the proposed schemes with the goal of verifying the claimed security and privacy properties. The relationship of the proposed schemes to the prior art is discussed in Section 9.9. Finally, the chapter concludes with a summary in Section 9.10.

9.2 Threat Model

The threat model underlying the schemes proposed in this chapter is similar to the threat model described in Section 6.2; that is, we address the threat of disclosure of the IMSI on the air interface. We follow the same approach to mitigating this threat, that is, to reduce the impact of IMSI disclosure, thereby enhancing user privacy. In doing so we make use of pseudonymous IMSIs.

In designing the schemes we make the underlying assumption that the 3G and 4G AKA protocols are sound, and provide mutually authenticated key establishment. We also implicitly assume that the USIM, the ME, and the network (both the SN and HN) have not been compromised by other means. Of course, if these assumptions are false, then very serious threats exist to both user privacy and security. The main risk introduced by use of the pseudonymous IMSIs is the possibility of loss of IMSI synchronisation between UE and HN, and this issue is addressed in Sections 9.5.3 and 9.7.3.

The schemes rely on using RAND hijacking, as described in Section 5.3. From our assumption regarding the security of AKA, we can assume that RAND hijacking provides an authenticated channel with replay detection. This is fundamental to the schemes presented in this chapter.

The schemes are designed to combat active adversaries, as described in the adversary model of Section 4.3. For the schemes described in Sections 9.3 and 9.4, we assume that an adversary has no control over the core network entities involved in communications, and communications across the core network are secured. However, in the robust pseudo-IMSIs scheme described in Section 9.6, we assume that the adversary may have access to selective core network functionalities (see Section 9.5.7).

9.3 Predefined Multiple IMSIs

We introduced this scheme in the context of GSM in Section 6.5, where we described how a fixed set of IMSIs for a user can be used to increase pseudonymity across the air interface. We introduced two approaches to triggering IMSI changes, namely SIM-initiated and network-initiated. The SIM-initiated approach can be used in 3G and 4G in the same way as described in Section 6.5.1. In this section we describe how the network-initiated approach can be used in 3G and 4G.

9.3.1 Protocol Operation

As discussed in Section 6.3, when the HN decides to trigger an IMSI change, it must, by some means, send an instruction to the USIM. We propose to use the AKA protocol as the communications channel for this instruction. More specifically, we propose using the value *RAND* of AKA to carry the signal; that is, we propose to make use of RAND hijacking, as described in Section 5.3. The IMSI change procedure operates as follows (see also Figure 9.1).

1. When the logic in the HN decides that an IMSI change is necessary, a flag is set for the appropriate user account in the AuC database of the HN.

2. Whenever the AuC generates authentication vectors for use in AKA, it checks this flag to see if an IMSI change signal is to be embedded in the *RAND* value. If so, it resets the flag and executes the following steps (as in Figure 9.1(a)).

   (a) The AuC uses the MAC function $f1$[2] to generate a 64-bit *MAC* on the subscriber's current sequence number *SQN* using the subscriber's long term key K. We refer to this as the *sequence-MAC* or *SMAC*. The *SMAC* is used to instruct the USIM to change its IMSI, i.e. it functions as an IMSI change signal. The reason the *SMAC* is 64-bits long is because we propose to compute it using an existing function; however, a value of a different length could be used, as long as it fits within the 128-bit *RAND*.

   (b) The AuC generates a 64-bit random number R using the same process as normally used to generate 128-bit *RAND* values.

   (c) The AuC sets *RAND* to be the concatenation of the R and *SMAC*.

[2]For cryptographic cleanliness it should be ensured that the data string input for this additional use of $f1$ can never be the same as the data string input to $f1$ for its other uses; alternatively, a slight variant of $f1$ could be employed here.

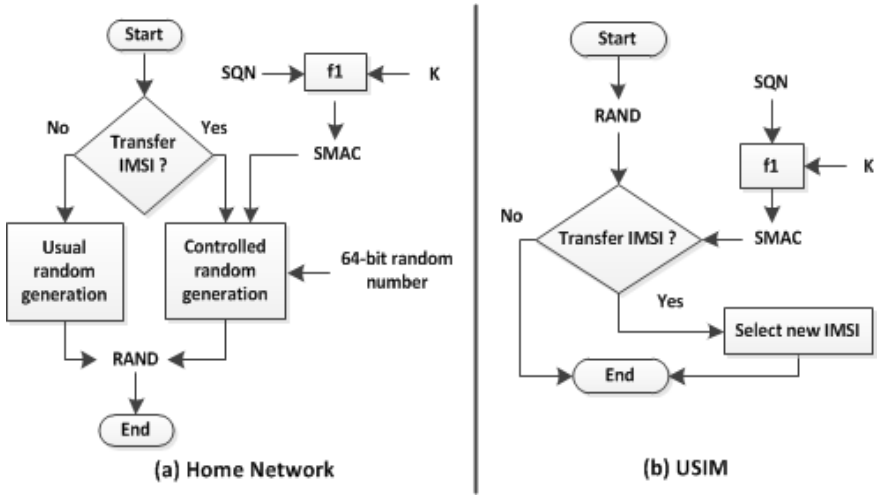

Figure 9.1: Additional computations for predefined multiple IMSIs

If an IMSI change signal is not required, the AuC generates *RAND* in the normal way.

3. The AuC follows the standard steps to generate the authentication vector from *RAND*, and sends the vector (including *RAND*) to the SN.

Whenever the USIM receives an authentication request, it follows the usual AKA steps. If the AKA procedure completes successfully, the USIM checks the *RAND* in the following way (as shown in Figure 9.1(b)).

1. The USIM uses the received *SQN* and its stored key K to regenerate *SMAC*.
2. It compares the computed *SMAC* with the appropriate part of *RAND*.
3. If they do not agree then the USIM terminates the checking process. However, if they agree then the USIM performs the next step.
4. The USIM selects a 'new' IMSI value from the stored list in the same way as described in Section 6.5.1, and later changes the IMSI using the procedure described in Section 6.4.

9.3.2 Discussion

We now consider how IMSI changes will work in practice. There are two cases to consider. If the HN is also the SN then it could potentially force an instance of AKA to occur at will, i.e. making the IMSI change happen almost immediately. However, if

the SN is distinct from the HN, then the HN can only send new AVs when requested by the SN. Moreover, the SN may delay before using the supplied AV in AKA. That is, there may be a significant delay between the decision being made to change an IMSI and the signal being sent to the USIM. In either case the phone may be switched off or temporarily out of range of a base station, in which case there will inevitably be some delay. However, regardless of the length of the delay in the signal reaching the USIM (or even if it never reaches the USIM) there is no danger of loss of IMSI synchronisation between the USIM and the HN, since the HN will always keep the complete list of IMSIs allocated to the USIM.

We observe that there is always the chance that a randomly chosen *RAND* will contain the 'correct' *SMAC*, leading to an unscheduled IMSI change by the USIM. However, the probability of this occurring is 2^{-64}, which is vanishingly small. In any case, the occurrence of such an event would not have an adverse impact, since the HN would always be aware of the link between the new IMSI and the particular USIM.

An active interceptor could introduce its own *RAND* into the channel to try to force an IMSI change. However, given that K is not compromised and $f1$ has the properties required of a good MAC function, then no strategy better than generating a random *RAND* will be available. Replays of old *RAND* values will be detected and rejected as a normal part of AKA, which enable the USIM to check the freshness of an authentication request. Also, assuming the *SMAC* value is indistinguishable from a random value, a standard assumption for MAC functions, then an eavesdropper will be unable to determine when an IMSI change is being requested.

Finally, we briefly observe how this scheme could be combined with the GSM RAND hijacking approach described in Chapter 5. In the Chapter 5 scheme, the *RAND* value in an AV contains a 16-bit field, known as the *AMF*. This *AMF* value could be used to transfer the IMSI change instruction, analogously to the scheme described above. Since the AMF value in the modified GSM scheme is confidentiality and integrity protected (see Section 5.8.3), such steps will ensure that the security and privacy properties are maintained.

9.4 Modifiable Multiple IMSIs

As described in Section 6.6, this scheme involves distributing new IMSI values from the HN to the USIM after its deployment, where the HN will choose each new IMSI from its pool of unused values. Such an approach clearly requires a means of communicating from the HN directly to the USIM. Analogously to the scheme proposed in Section 9.3.1, we describe how RAND hijacking can be used for this purpose.

9.4.1 Prerequisites

Before describing the details of the IMSI transfer procedure, we describe below some relatively minor changes which are required to the operation of the HN in order to support the scheme.

- The HN must maintain a pool of unused IMSIs, enabling the AuC to dynamically assign a new IMSI to an existing subscriber.
- For each subscriber account in its database, the HN must maintain an *IMSI-change flag* indicating whether an IMSI change is under way. The database must also hold up to two IMSIs for each subscriber; it will always hold the current IMSI (with status *allocated*) and, if the *IMSI-change flag* is set, it will also hold the new IMSI (with status *in transit*), where the possible status values for an IMSI are discussed below. If use of the new IMSI is observed then IMSI status changes are triggered (see below).
- The HN must manage the use of IMSIs so that no IMSI is assigned to more than one subscriber at any one time. This can be achieved by maintaining the status of each IMSI as one of *allocated*, *free*, or *in transit*. The set of IMSIs with status *free* corresponds to the pool of available IMSIs, as above. The status of an IMSI can be updated in the following ways.
  - When the HN selects an available IMSI from the pool to allocate to a USIM, the status is changed from *free* to *in transit*.
  - When the HN receives implicit acknowledgement (in the form of a request for AVS for that IMSI from a network) of a successful IMSI change, the HN changes the status of the IMSI from *in transit* to *allocated*. The HN also changes the status of the previously used IMSI for that subscriber from *allocated* to *free* after a delay (to avoid potential collisions in network use). In addition, the current IMSI for the subscriber will be set equal to the new IMSI, the new IMSI will be set to null, and the *IMSI-change flag* will be reset. This will happen when the IMSI in the request message for AVs agrees with an IMSI with status *in transit* in the subscriber database. Otherwise (i.e. when the IMSI in the request message for AVs agrees with an IMSI with status *allocated*), the HN computes the AVs according to the procedure described in Section 9.4.2 below.
  - A third case also needs to be considered, that is when an IMSI change instruction never reaches the USIM. If this case is not addressed then future

IMSI changes for that USIM will be blocked. On the other hand, making a decision to abandon an IMSI change could be disastrous, i.e. if a USIM makes an IMSI change after the HN has terminated this change (and changed the status of the 'new' IMSI back to *free*), then the USIM could be rendered unusable. As a result we propose never to abandon an IMSI change, and instead to resend the new IMSI as many times as necessary until the change is accepted by the USIM. How this works should be clear from the protocol operation described in Section 9.4.2 below.

- If the HN is required to do so by its regulatory environment, e.g. to support lawful interception, it can maintain a log of all the IMSIs assigned to a particular subscriber for however long is required. It is in any case likely to be necessary to retain this information for a period to enable processing of billing records received from visited networks.

9.4.2 Protocol Operation

The scheme introduces changing IMSIs. The new IMSI is used in exactly the same way as a regular IMSI, i.e. when polled for its IMSI the phone will respond with its new IMSI. The operation of AKA (see Figure 9.2) is also unchanged; the only difference is in the composition of *RAND*, as discussed below, and this difference is transparent to the SN.

The IMSI transfer procedure requires changes in the HN and the USIM. We now describe the necessary changes to the operation of an HN in computing the AV.

1. When the logic in the HN decides that an IMSI transfer is necessary for a particular subscriber, it must set the *IMSI-change flag* for that subscriber. Observe that if an IMSI change is already under way then the flag will already be set; in this case the flag is left as it is.

2. Whenever the AuC needs to generate AVs for use in AKA, it checks this flag to see whether an IMSI transfer signal and a new IMSI are to be embedded in the *RAND* value. If so, it performs the following steps (as shown in Figure 9.3). Note that this means that, once an IMSI change has been initiated, the new IMSI will be embedded in all *RAND* values until evidence of the successful changeover by the USIM has been observed.

   (a) The AuC uses the MAC function $f1$ to generate a 64-bit *MAC* on the subscriber's current sequence number *SQN* using the subscriber's long term

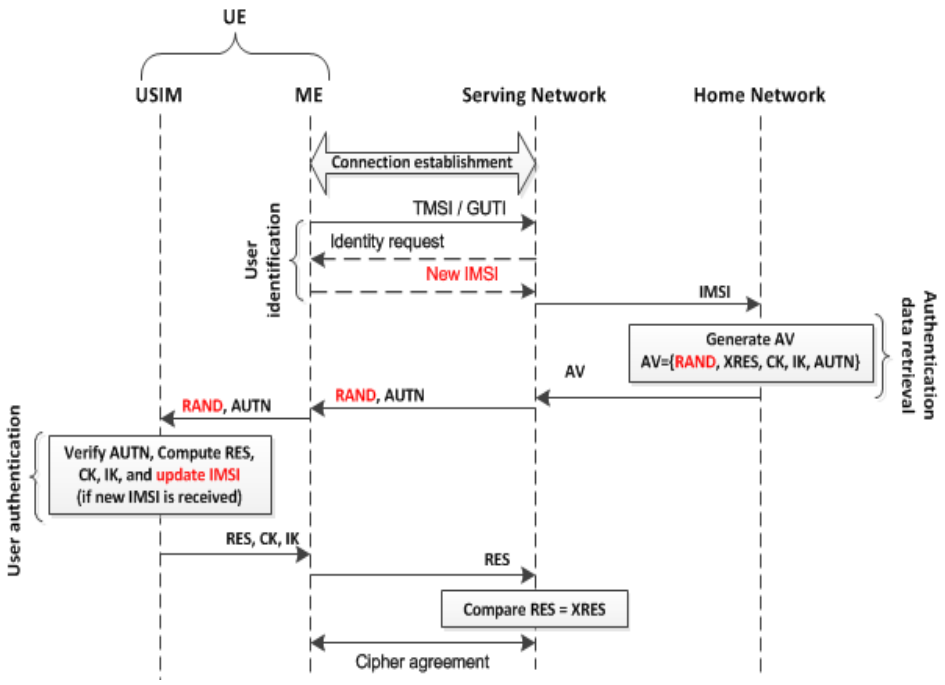

Figure 9.2: Authentication message flow for modifiable multiple IMSIs

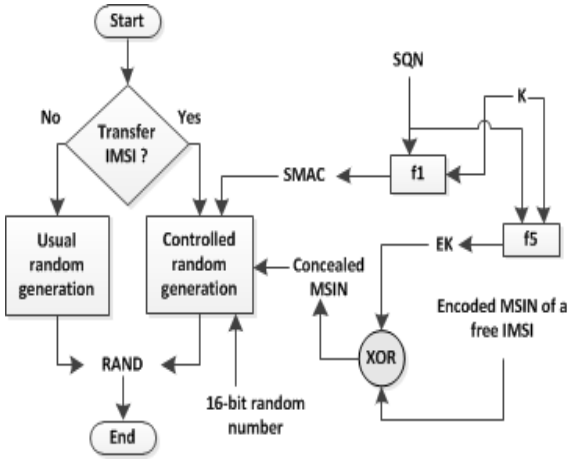

Figure 9.3: Computations in AuC for modifiable multiple IMSIs

cryptographic key K. As in the predefined multiple IMSIs scheme described in Section 9.3.1, the generated MAC is referred to as the $SMAC$.

(b) The AuC generates a 48-bit encryption key EK using the key generation function $f5$. The function takes SQN as the data input and K as the key input. Note that observations regarding cryptographic cleanliness and the use here of functions $f1$ and $f5$, analogous to those given in Section 9.3.1

step 2a, apply here.

(c) If the new IMSI field in the HN database entry for this subscriber is non-null then a new IMSI has already been assigned, and it is not necessary to choose another new value. Otherwise a new IMSI is selected from the pool of unused IMSIs; the status of this IMSI is changed from *free* to *in transit* and the new IMSI field in the database is given the chosen value. We assume that the MCC and MNC of the IMSI are known to the USIM (since they are fixed for this network operator) and hence only the 9- or 10-digit MSIN needs to be sent embedded in $RAND$. The MSIN is encoded as a 36- or 40-bit value using binary coded decimal, the 'standard' way of encoding IMSIs, and the result is padded to 48 bits by an agreed padding scheme.

(d) The 48-bit MSIN block is XORed with the encryption key EK, and we refer to the result as the concealed MSIN.

(e) The AuC generates a 16-bit random number R using the same process as normally used to generate 128-bit $RAND$ values.

(f) The AuC sets $RAND$ to be the concatenation of the concealed MSIN, R and $SMAC$.

If an IMSI transfer is not required, the AuC generates $RAND$ in the normal way.

3. The AuC follows the standard steps to generate the AV from the $RAND$ value, and sends it (including $RAND$) to the SN.

We now describe in detail the necessary changes to a USIM to retrieve the new IMSI and to transfer the IMSI to an ME. On receipt of an authentication request, the USIM proceeds using the standard AKA procedure. After successful completion of the AKA protocol, the USIM checks whether the challenge value contains an embedded IMSI in the following way (as shown in Figure 9.4).

1. The USIM uses the received SQN and its stored long term key K to regenerate $SMAC$.

2. It compares the computed $SMAC$ with the appropriate part of $RAND$.

3. If they do not agree then the USIM terminates the checking process. However, if they agree then the USIM performs the following steps.

   (a) The USIM retrieves the concealed MSIN from $RAND$.

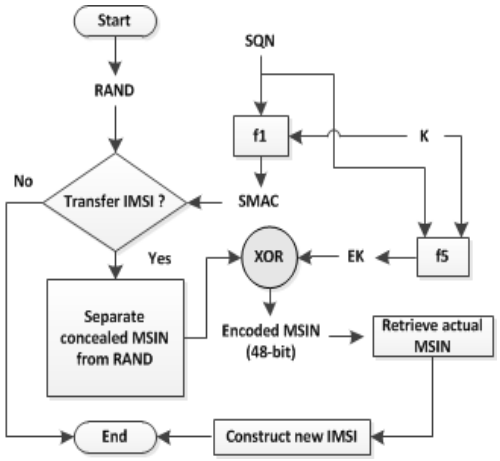

Figure 9.4: Computations in USIM for modifiable multiple IMSIs

(b) The USIM regenerates the encryption key EK using $f5$ with the value of SQN retrieved during the AKA processing and its long-term stored key K as inputs.

(c) The EK is XORed with the concealed MSIN to recover the cleartext BCD-encoded MSIN.

(d) The USIM generates the new IMSI by prefixing the decoded MSIN with the MCC and MNC.

(e) The USIM checks whether the new IMSI is the same as the value it is using already; this is essential since it may receive the change instruction more than once. If they are the same it takes no further action. If they are different it keeps a record of the new IMSI and later updates its IMSI using the procedure described in Section 6.4.

To reduce signalling costs, it appears to be standard practice for the AuC to generate a small set of AVs for provision to an SN. If the procedure specified above is followed to generate this set of vectors, and an IMSI change is scheduled for the subscriber, then all the $RAND$ values in the set will contain an embedded concealed MSIN. Whilst this will cause minimal additional overhead for the USIM, since $RAND$ values are always checked for an embedded $SMAC$ value, it will have the benefit of maximising the chance that the IMSI change will be performed by the USIM.

9.5 Analysis of Modifiable Multiple IMSIs

9.5.1 Correctness of the Scheme

The scheme inherits its security strength from the AKA protocol as it does not modify the existing AKA except for the method used to generate *RAND*. The modified AKA protocol is thus as secure as the existing AKA. The security properties of AKA have been widely analysed, [1, 120, 155]. We formally verified this claim using ProVerif (see Section 9.8.1).

The computed *RAND* is constructed so that it is indistinguishable from a random value; this claim is based on the assumption that the output of the $f1$ variant function and a data string masked using the output of the $f5$ variant function are indistinguishable from random data, cf. [18, 19]. Use of the varying *SQN* in the scheme randomises the MSIN-carrying *RAND* when the same pseudonym is sent in multiple challenges.

The scheme does not change the way the data confidentiality and integrity key are generated, and so the strength of cryptographic key generation is not affected.

The scheme achieves the following two security goals:

1. the subsequent IMSIs are unlinkable by the air interface adversary; and
2. an IMSI used by a USIM is always the one that was previously communicated by the HN, i.e. the HN expects every sent pseudonymous IMSI to be used.

We next give more detailed arguments supporting these two claims.

9.5.1.1 Goal 1: IMSI Unlinkability

IMSI unlinkability depends on the following two assumptions.

1. pseudonymous IMSIs are randomly picked from an extensive list; and
2. a pseudonymous IMSI (actually the MSIN part of an IMSI) is confidentially communicated from the HN to the USIM.

Of these, assumption 1 is an implementation issue, whereas assumption 2 is dependent on the confidentiality method in use. Section 9.4.2 describes how a newly selected pseudonym, i.e. the MSIN part of a successor IMSI, is sent to a USIM. The confidentiality mechanism involves masking the pseudonym by XORing it with a pseudorandom sequence output by $f5$. That is, MSIN confidentiality relies on the effectiveness of a function already present in a standard USIM. An active or passive adversary without access to the shared secret key K is therefore unable to learn the new IMSI before it is used, ensuring unlinkability between consecutive IMSIs. We formally verified the confidentiality property of the transferred MSIN, using ProVerif (see Section 9.8.1).

9.5.1.2 Goal 2: IMSI Correctness

Achieving IMSI correctness follows from the security properties of AKA. As the new pseudonym for a USIM is communicated through the use of AKA, the USIM will only switch to a new pseudonym if it was communicated inside an authentic and fresh 'authentication request' message. That is, as a by-product of network authentication, AKA guarantees the origin, integrity and the timeliness of the *RAND* value containing the new pseudonym. Hence, an active adversary cannot force the USIM to change its IMSI to something other than a value selected by the HN.

If the 'authentication request' message contains an 'old' *SQN* value, then the included pseudonym will be ignored by the USIM, which means it will not update its stored current IMSI. The scheme thus protects against attempts to force the USIM to switch to an 'old' pseudonymous IMSI.

The HN will only accept a switch to a successor pseudonym after receiving a request for authentication vectors containing the successor pseudonym, which should always be sent when a new IMSI attaches. Thus even an active attacker cannot force a USIM to accept a pseudonymous MSIN other then the one expected by the HN.

9.5.2 User Privacy

As discussed in Section 6.8, the scheme does not provide a complete solution to user identity confidentiality. However, it diminishes the impact of IMSI catchers and improves user identity confidentiality by reducing the effect of IMSI disclosure across the air interface. Since the IMSI functions as a pseudonym, air interface interactions are not completely anonymous, potentially enabling the interactions of a single subscriber to be tracked for a period. However, frequent IMSI changes could lessen the impact of such tracking.

As discussed in Section 9.5.1.1, the design of the scheme ensures an eavesdropper is unable to infer any confidential information from the value of *RAND*. We formally verified the confidentiality property of the private data embedded in *RAND* using ProVerif (see Section 9.8.1).

As discussed in Section 9.5.1, the *RAND* is constructed so that it is indistinguishable from a random value. Thus an eavesdropper cannot differentiate between an IMSI-changing AKA execution from a standard AKA interaction.

9.5.3 IMSI Synchronisation

As discussed in Section 6.8, loss of IMSI synchronisation appears to be a significant threat; therefore, robust means are necessary to guarantee the correctness and timeli-

ness of the IMSI held by a USIM.

As discussed in Section 9.3.2, there may be a significant delay in the IMSI change signal reaching the USIM; however, this will not affect IMSI synchronisation between the HN and the USIM since the HN will not update the current IMSI entry in the subscriber database until it receives a request for authentication vectors from an SN using this new IMSI.

As discussed in Section 9.4.2, once a new IMSI has been assigned to a subscriber (with the *in transit* status), every *RAND* generated for that USIM will contain the embedded IMSI value until success of the change has been observed; that is, the loss of authentication data containing an IMSI-embedded *RAND* will not affect IMSI synchronisation between the HN and the USIM.

As discussed in Section 9.5.1.2, a USIM accepts a new IMSI sent embedded in a *RAND* only after AKA has completed successfully, i.e. after the network has been authenticated. Hence, an active adversary cannot force the USIM to change its IMSI to a value chosen by the adversary. Further, because *SQN* is checked by the USIM during AKA, an active adversary cannot force a USIM to accept an 'old' IMSI, since this checking forces AVs to be used in strict order of generation. Moreover, malicious changes to a valid *RAND*, e.g. involving changing the encrypted MSIN whilst leaving the *SMAC* unchanged, will be detected by the AKA network authentication process. These security features combine to help ensure that IMSI synchronisation is preserved.

In the above discussion we assume that SNs always behave honestly and that communications between networks is secure; therefore, IMSI synchronisation is maintained as long as the assumptions hold. However, if the assumptions do not hold, IMSI desynchronisation could arise in a variety of ways.

One possible scenario involves a compromised SN sending randomly chosen IMSIs to an HN, e.g. embedded in an AV request. If the received IMSI is equal to a currently free IMSI, the HN will respond with an error in the standard way. If the IMSI is equal to a stored current IMSI (an $IMSI_{allocated}$), the HN will provide an AV. Finally, and most importantly here, if the IMSI is equal to a stored future IMSI (an $IMSI_{intransit}$), then there will be a loss of IMSI synchronisation if the mobile to which the stored future IMSI belongs has not received it. This could cause a permanent denial of service, since there will be no way for synchronisation to be regained. In Section 9.6 below, we describe an enhanced version of the modifiable multiple IMSIs scheme that incorporates an identity synchronisation recovery process, designed to address this threat.

A related but distinct scenario involves a malicious or malfunctioning ME submitting random IMSIs to an SN. The SN will use a received IMSI to request an AV from the HN indicated by the PLMN-ID part of the IMSI. The remainder of the attack

will work exactly as in the scenario described in the previous paragraph; that is, if the random IMSI happens to equal an $IMSI_{intransit}$ for a USIM which has not yet received this IMSI, then the corresponding USIM will suffer from a catastrophic identity desynchronisation.

The IMSI change instruction in this scheme has similarities to that of the predefined multiple IMSIs scheme described in Section 9.3. As in Section 9.3.2, there is the chance that a randomly chosen *RAND* could contain a 'correct' *SMAC*, triggering an unauthorised IMSI change. However, for similar arguments to those given in Section 9.3.2, the probability of such an event is vanishingly small, and certainly orders of magnitude smaller than the probability of a USIM failure.

9.5.4 Performance and Overhead

The scheme introduces minimum overhead for a USIM. We add one MAC function (to decode the IMSI change instruction) and one key generating function (to retrieve a new IMSI), both of which are similar to the existing USIM functions. Transferring a new IMSI to the ME is a new task for a USIM, which could be performed when the ME is in idle state. We believe that this overhead should be manageable, even for a USIM with limited computational power.

The scheme adds overhead to the HN for managing multiple IMSIs for a subscriber. It adds new database transactions, e.g. to update an IMSI. It adds two cryptographic functions for computing an AV, and introduces a new function to refresh an IMSI. Since none of these are particularly complex, it seems likely that this could be achieved with some combination of allocating more resources, clustering subscribers in multiple HLRs/HSSs, and efficient database design.

The scheme does not affect any operations in the SN or introduce any additional communications. The only impact is an increase in the apparent number of subscribers at the SN, since subscribers switching to a new IMSI appear like new subscribers.

Since multiple IMSIs are allocated for each subscriber, pressure could be created on the number of IMSIs available to an operator. To address this issue, multiple MNC codes could be allocated to an operator, and three-digit MNCs could be used to avoid wastage of IMSIs allocated to small operators.

9.5.5 Deployment and Interoperability

The scheme modifies only the USIM and the HN, which are owned by a single entity, and is transparent to the intermediate SN and mobile phone. This allows phased deployment, e.g. by including the additional functionality in newly issued USIMs while

existing USIMs continue to function as at present.

There are certain practical issues to be considered. For example, the set of 'normal' IMSIs used by existing USIMs needs to be kept distinct from the range of changing IMSIs used by the new USIMs; however, choosing them from completely distinct ranges would not be desirable since it would enable them to be distinguished.

The scheme will not work for GSM, as it depends on the mutual authentication feature of 3G and 4G AKA. If a UE using a new-style USIM needs to connect to a GSM network, it should continue to use the current IMSI as long as it is connected to that network. The IMSI can be updated when the UE next roams to a 3G or 4G network.

9.5.6 A Related Scheme

As mentioned in Section 6.6, a related scheme, developed in parallel by van den Broek, Verdult and de Ruiter [163], proposes an approach similar to the modifiable multiple IMSIs scheme described in Section 9.4. We next describe this scheme.

The subscriber's IMSI is replaced with a changing pseudonym, the PMSI, which is only resolvable by the USIM's HN. The structure of a PMSI is the same as that of an IMSI, and it is treated like an IMSI by the SN which is unaware of the fact it is not an IMSI. The scheme requires the AuC to store three additional values for each USIM: a new shared secret key K_e, the current PMSI in use by the subscriber, and a future PMSI. When requested by an SN for an AV for a particular PMSI, the AuC executes the following steps (see Figure 9.5(a)).

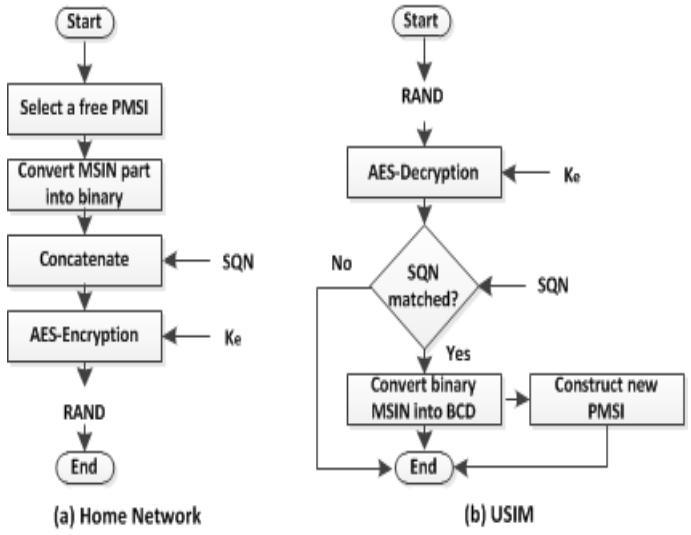

Figure 9.5: Additional computations for the van den Broek et al. scheme [163]

1. It compares the PMSI with all current and future PMSIs, and continues if it finds a match.

2. If the match is with a future PMSI, the AuC updates the subscriber's current PMSI to equal the received PMSI, and assigns a randomly chosen PMSI from the pool of free PMSIs to be the subscriber's future PMSI. If the match is with a current PMSI then the AuC database is unchanged. Note that only the MSIN part of the PMSI is actually sent, encoded as a 34-bit string by taking the (at most) 10 decimal-digit MSIN, treating it as a number, and using its binary representation.

3. The AuC computes the $RAND$ by AES-encrypting the concatenation of the 34-bit binary-coded MSIN part of the future PMSI and the subscriber's current SQN, using the key K_e.

4. The AuC computes the other authentication parameters in the AV as a function of the computed $RAND$ (in the standard way), and transmits the AV to the SN.

The USIM must possess K_e, along with the current and future PMSIs. On receiving an authentication challenge ($RAND$ and $AUTN$), the USIM executes the usual AKA steps; if AKA is successful, the USIM decrypts the received $RAND$ and compares the recovered SQN value with the received SQN. If they agree, the USIM extracts the binary-encoded MSIN, converts it to binary coded decimal, and uses it to create the new stored future PMSI. The process is illustrated in Figure 9.5(b).

The USIM continues for the moment to use its current PMSI. When the USIM next receives an identity request, it sets its stored current PMSI to the stored future PMSI, and responds with this new value. That is, an updated PMSI is only used on the next occasion that the USIM receives an identity request, preventing linking of new and old PMSI values by observers. However, how the ME will be made aware of the new PMSI is not discussed.

9.5.7 Practical Issues

In the modifiable multiple IMSIs scheme discussed in Section 9.4.2, network operators selectively change subscriber IMSIs, embedding a USIM-specific value in $RAND$ to instruct the USIM to change its IMSI. The van den Broek et al. scheme, described in Section 9.5.6, uses a similar RAND-embedded value, but changes PMSIs at every authentication. Both schemes involve updating the IMSI held by the HN as soon as use of the new IMSI by the mobile is observed.

In both schemes, and as discussed in Section 9.5.3, it is implicitly assumed that SNs always behave honestly and that communications between networks is secure. However, security flaws in the core network protocols and their implementation have recently been widely discussed, [57, 77, 80, 95, 97, 133, 139, 140, 161]. Bilogrevic, Jadliwala and Hubaux [57] and Golde, Redon and Borgaonkar [97] describe possible vulnerabilities in the mobile core network in the presence of femtocell, a miniature cellular base station widely deployed in LTE networks. Rao et al. [139, 140] describe attacks exploiting flaws in the *signalling system 7* (SS7) and *diameter* protocols, widely used in core network communications. Nohl [133] and Engel [80] independently demonstrate flaws in the SS7 protocol. This means that the assumptions underlying the modifiable multiple IMSIs and van den Broek et al. schemes are questionable, meaning that the possibility of loss of identity synchronisation needs to be addressed.

Also, both the schemes involve changing the IMSI at the HN. This introduces major management concerns because the IMSI is the only unique identifier in the HN, and other services, for example the IP multimedia subsystem (IMS), depends on fixed IMSIs. Thus, a scheme supporting pseudonyms without changing the IMSIs in the HN would be highly advantageous.

These observations have motivated the development of an enhanced pseudonym-based scheme that can provide pseudonymity across the air interface, allows recovery from loss of identity synchronisation, in a way that minimises changes to the deployed network infrastructure. We describe such a scheme in the next section.

9.6 Robust Pseudo-IMSIs

We now describe a new set of modifications to the operation of 3G and 4G systems designed to address the same privacy concerns as the modifiable multiple IMSIs scheme, and using very similar ideas. However, it incorporates significant additional features that protect against permanent loss of identity synchronisation between a USIM and its HN. The scheme also avoids changing the IMSI in the HN, i.e. it addresses both the major concerns raised in Section 9.5.7.

9.6.1 Overview

Just like modifiable multiple IMSIs, the scheme uses changing pseudonyms. The scheme avoids sending the subscriber IMSI across any communication channels, including the radio path and the core network, and instead uses a *transient identity* (TID), i.e. a temporary identity managed by the HN with the same length as an MSIN. The TID works rather like the TMSI except that TIDs are managed by the HN instead of the

SN. The TID provides subscriber pseudonymity on the air interface, avoiding the need for cleartext transmission of the IMSI. TIDs are mapped to fixed IMSIs in the HN database, and the IMSI is only accessible by the HN.

The initial pseudonym (the *pseudo-IMSI*), equal to the concatenation of the PLMN-ID and the initial TID, must be stored in a USIM during personalisation. The pseudo-IMSI is indistinguishable from an IMSI to any party other than the USIM and HN. As in modifiable multiple IMSIs, the pseudo-IMSI is treated as an IMSI by the SN, and is periodically refreshed by the HN. An unauthenticated UE (without a valid TMSI/GUTI) identifies itself to the SN using the pseudo-IMSI, exactly as an IMSI is used currently. The SN learns the identity of the HN from the pseudo-IMSI, and forwards it to the HN as part of an AV request. The HN uses the TID from the pseudo-IMSI to learn the IMSI. It then chooses a new TID to refresh the pseudo-IMSI, embeds this new TID into the *RAND* analogously to modifiable multiple IMSIs, and computes the AV, which is sent to the SN for use in AKA.

After a successful AKA procedure, the USIM has the new TID, and uses it to construct the next pseudo-IMSI. At the same time, the SN allocates the UE a local TMSI or GUTI. As in modifiable multiple IMSIs, the old pseudo-IMSI remains in use by the current SN, e.g. in mobile terminated services (if required). The new pseudo-IMSI is used in subsequent AKA executions. We also introduce the notion of a *recovery identity* (RID) for each USIM, to enable pseudo-IMSI recovery in a privacy-preserving way.

9.6.2 Modifications to AKA

The pseudo-IMSI is indistinguishable from an IMSI to the SN, and is used in exactly the same way, i.e. when polled for its IMSI the phone will respond with its pseudo-IMSI. The operation of AKA (see Figure 9.6) is also unchanged; the only difference is in the composition of *RAND*, as discussed in Sections 9.6.3 and 9.6.4, and this difference is transparent to the SN.

The only other change of relevance here is in error handling. As described in Section 3.4.2, if a USIM fails to authenticate a network, it responds with either a *MAC-failure* message (sent via the signalling channel), or a *sync-failure* message (sent in a failure token that is forwarded by the SN to the HN). As part of the novel scheme (discussed in greater detail in Section 9.6.4.2) if the authentication fails due to a MAC mismatch, the USIM incorporates a novel failure type indication in an error token that is sent to the SN (and hence to the HN) just like a *sync-failure* error. Since the SN does not process the error token, the incorporation of a new type of error code is transparent to it.

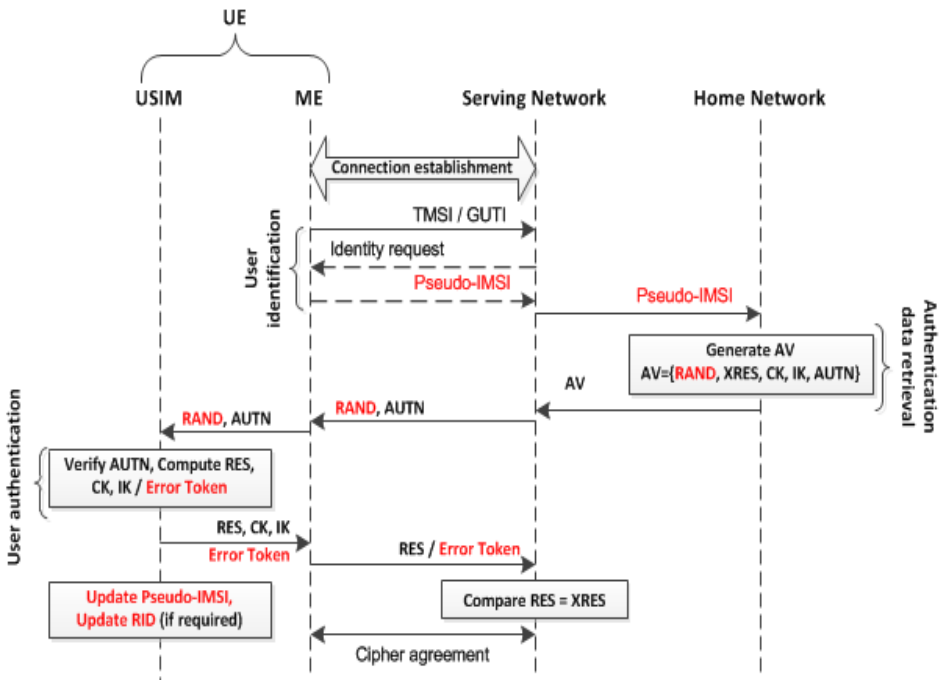

Figure 9.6: Authentication message flow for robust pseudo-IMSIs

That is, the only change from the SN perspective is that it will never see a *MAC-failure* message; a malicious third party could use this to distinguish between an IMSI and a pseudo-IMSI by sending a modified (and hence incorrect) MAC value to a UE and seeing what error message is returned.

9.6.3 Modifications to Home Network

We now describe in detail the necessary changes to the HN processes.

9.6.3.1 Fundamentals

We first introduce certain data structures which form part of the modified scheme.

- *TID*: A TID substitutes for the MSIN, and so its maximum length is 40 bits (i.e. ten binary coded decimal digits). If the HN implementation encodes the HSS/HLR identity in the MSIN, as is done in some networks, the bit-length of a TID could equal the number of bits used to uniquely identify a subscriber of that specific HSS/HLR.
- *RID*: Like the TID, a RID is a 48-bit USIM identifier, used only in the pseudo-IMSI recovery procedure described in Section 9.6.5.2.

- *Linked TID*: This is a specific TID, sent with a RID to a subscriber. This TID (if not null) is equal to the value of TID_{future} stored by the HN (see below).
- *RID Flag*: This bit, maintained for every subscriber in the HN database, indicates whether or not a RID value should be embedded into the next *RAND* for this subscriber. Initially the flag is cleared. The use of the flag is similar to that of the *IMSI-change flag* in modifiable multiple IMSIs.
- *Keystream 1 (EK_1)*: Like *EK* in modifiable multiple IMSIs, the 48-bit EK_1 is derived from subscriber-specific secrets using a key derivation function. It is used to mask the TID when sent from the HN to a USIM.
- *Keystream 2 (EK_2)*: This is another 48-bit string generated using a different KDF from subscriber-specific secrets. It is used to mask the RID when it is transferred from the HN to a USIM.
- *Padding data*: Pad_1 and Pad_2 (each of 80 bits) are used to extend the subscriber's *SQN* value to a 128-bit string, as required for input to the $f5$ variant functions.
- *Instruction byte*: This is used to instruct a USIM to perform certain operations. We use specific values of the byte ($INS_{regular}$, INS_{RID} and INS_{reset}) to instruct the USIM to perform certain tasks. All other values are reserved for future use, e.g. to give the USIM other types of instruction — on receipt of a reserved value the USIM should not take any action.

For each subscriber account in the database, the HN must maintain a *RID flag* indicating whether a RID change is under way. The database must also hold up to three TIDs (TID_{past}, $TID_{current}$ and TID_{future}), three RIDs (RID_{past}, $RID_{current}$ and RID_{future}) and a *linked TID* for each subscriber. It will always hold a $TID_{current}$ and a $RID_{current}$. When a new TID is transferred to the USIM, the database will hold a TID_{future}; after a TID update, described in Section 9.6.3.3, it will hold a TID_{past}. If the *RID flag* is set, the database will also hold a *linked TID* and a RID_{future}. Analogously to the TID, after a RID update, the database will hold a RID_{past}. The HN must also maintain a pool of unused TIDs and RIDs, enabling the AuC to dynamically assign a new TID or RID to an existing subscriber.

The following functions are used in the scheme described below.

- $f5^{**}$: This KDF is a variant of the existing $f5$. It takes as input a 128-bit string and the shared secret key K, and generates EK_1. The function must be chosen so that it is computationally infeasible to derive the key K from knowledge of the string and EK_1, i.e. exactly as is required of $f5$ as described in Section 3.4.3.

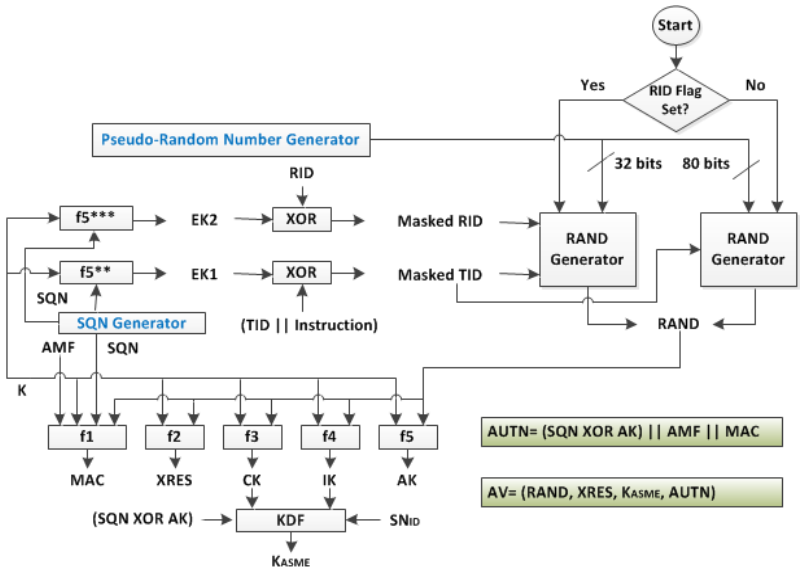

Figure 9.7: Computations in AuC for robust pseudo-IMSIs

- $f5^{***}$: This function, used to generate EK_2, must have similar properties to $f5^{**}$, but for cryptographic cleanliness we use a distinct function.
- *Randgen*: This function takes as input an integer, and returns a pseudo-random number of the length specified.

9.6.3.2 AV Generation

On receiving a request for an AV for a pseudo-IMSI, the AuC first retrieves the TID from the pseudo-IMSI and searches its subscriber database for the corresponding IMSI. If the IMSI is not found, the AuC generates an arbitrary AV in which all the values are chosen at random (to initiate pseudo-IMSI recovery by the receiving USIM, as described in Section 9.6.5.2), and sends it to the SN. Otherwise, the AuC retrieves the IMSI and proceeds as follows (see Figure 9.7). Algorithm 9.1 provides a pseudocode description of this process.

1. It retrieves the shared key K, SQN, RID flag, and stored value of TID_{future} for this IMSI.
2. It sets EK_1 to equal $f5^{**}(SQN \parallel Pad_1, K)$.
3. If the value of TID_{future} is null, the AuC sets it to equal a fresh TID selected from the pool of unused TIDs. The AuC also removes the allocated TID from the unused TID pool.

Algorithm 9.1: AV generation in robust pseudo-IMSIs

Data: Pseudo-IMSI
Result: An authentication vector

begin
  $RTID \longleftarrow$ *TID part of the Pseudo-IMSI*;
  if *(RTID is not in subscriber database)* **then**
    Generate an arbitrary AV;
    return;
  else
    Fetch relevant data (assume in Row structure) from subscriber database associated with RTID;
    $IMSI \longleftarrow Row.IMSI$; $K \longleftarrow Row.K$;
    $SQN \longleftarrow$ *Subscriber specific SQN*;
    $ESQN_1 \longleftarrow SQN \parallel Pad_1$; $EK_1 \longleftarrow f5^{**}(ESQN_1, K)$;
    $NTID \longleftarrow Row.TID_{future}$;
    if $NTID \neq \emptyset$ **then** $TID_{fresh} \longleftarrow NTID$;
    else
      $TID_{fresh} \longleftarrow$ *an unused TID*;
      Update the value of TID_{future} *with* TID_{fresh};
    $Flag \longleftarrow Row.RID\text{-}Flag$;
    if $Flag = 1$ **then** `/* Both TID and RID are embedded into RAND */`
      $NRID \longleftarrow Row.RID_{future}$;
      if $NRID \neq \emptyset$ **then** $RID_{fresh} \longleftarrow NRID$;
      else
        $RID_{fresh} \longleftarrow$ *an unused RID*;
        Update the value of RID_{future} *with* RID_{fresh};
      Update the value of Linked-TID with TID_{fresh};
      $Encoded\text{-}TID \longleftarrow TID_{fresh} \parallel INS_{RID}$;
      $Masked\text{-}TID \longleftarrow Encoded\text{-}TID \oplus EK_1$;
      $ESQN_2 \longleftarrow SQN \parallel Pad_2$; $EK_2 \longleftarrow f5^{***}(ESQN_2, K)$;
      $Masked\text{-}RID \longleftarrow RID_{fresh} \oplus EK_2$;
      $RAND \longleftarrow Masked\text{-}TID \parallel Masked\text{-}RID \parallel randgen(32)$;
    else `/* TID is embedded into RAND */`
      $Encoded\text{-}TID \longleftarrow TID_{fresh} \parallel INS_{regular}$;
      $Masked\text{-}TID \longleftarrow Encoded\text{-}TID \oplus EK_1$;
      $RAND \longleftarrow Masked\text{-}TID \parallel randgen(80)$;
    Generate AV using the RAND in the standard way;

4. If the RID flag is not set, the AuC sets *masked-TID* to $(TID_{future} \parallel INS_{regular}) \oplus EK_1$, and sets *RAND* to equal *masked-TID*$\|randgen(80)$. Otherwise, the AuC computes *RAND* as follows.

   (a) It sets EK_2 to equal $f5^{***}(SQN \parallel Pad_2, K)$, and *linked-TID* to equal TID_{future}.

(b) If the value of RID_{future} is null, the AuC sets it to equal a fresh RID selected from the pool of unused RIDs. The AuC also removes the RID from the pool of unused RIDs.

(c) It sets masked-TID to $(TID_{future} \parallel INS_{RID}) \oplus EK_1$, and the masked-RID to $RID_{future} \oplus EK_2$.

(d) It sets $RAND$ to equal $masked\text{-}TID \| masked\text{-}RID \| randgen(32)$.

5. The AuC generates the AV using the computed $RAND$ in the standard way.

9.6.3.3 Identity Update

Although an HN keeps the subscriber's security credentials to itself, it delegates authentication responsibility to an SN by passing it an AV; the HN is therefore not aware when a specific AV is used in AKA. Thus the HN does not have a direct means of knowing when a USIM receives a new TID. As discussed in Section 3.6.2, a *location update* request from an SN is preceded by a successful AKA; we therefore use the receipt by an HN of a *location update* request as implicit indication that a USIM has received the TID in the provided pseudo-IMSI; we use this to trigger a TID update. This approach differs from the schemes described in Sections 9.4 and 9.5.6, and, as we describe in the next paragraph, this change to some extent restricts unauthorised updates to the HN database.

The schemes described in Sections 9.4 and 9.5.6 both involve the HN updating its database as soon as evidence is seen of use of a new IMSI, i.e. when there is a request for an AV for this IMSI. However, in Section 9.5.3 we described a scenario in which a malicious ME could, by inserting attach requests to a network for randomly chosen IMSIs, trigger an unauthorised update to the database if one of the random IMSIs equalled a 'new IMSI' which had not yet reached the intended USIM. That is, both the modifiable multiple IMSIs and van den Broek et al. schemes are vulnerable to such an attack, which could cause permanent loss of synchronisation between a USIM and its HN. This attack is prevented if the HN database update is delayed until a *location update* request is received for the 'new IMSI', since, as described in Section 3.6.2, such a message is not sent by the SN until the SN has successfully authenticated the MS using this IMSI. Of course, a variant of the other attack scenario described in Section 9.5.3, in which a malicious or faulty SN submits *location update* requests for randomly chosen IMSIs is not prevented, but this appears less likely than the malicious ME attack.

When an HSS receives a *location update* request it sends the embedded pseudo-IMSI to the AuC for a possible identity update. (In 3G, the *location update* request is sent

to the HLR and not the AuC although, since they are controlled by the same network, adding the necessary intercommunication should not be difficult.)

Algorithm 9.2: Identity update in robust pseudo-IMSIs

```
   Data: Pseudo-IMSI
   Result: Updated identities in the subscriber database
 1 begin
 2     RTID ⟵ TID part of the Pseudo-IMSI;
 3     Fetch relevant data (assume in Row structure) from subscriber database
       associated with RTID;
 4     NTID ⟵ Row.TID_future;
 5     if RTID = NTID then          /* Acknowledgement of a sent TID */
 6         CTID ⟵ Row.TID_current;
 7         OTID ⟵ Row.TID_past;
 8         Flag ⟵ Row.RID-Flag;
 9         Update the value of TID_current with NTID;
10         Update the value of TID_past with CTID;
11         Set TID_future = Ø;
12         Add OTID to the unused TID pool;
13         if Flag = 1 then
14             NRID ⟵ Row.RID_future;
15             CRID ⟵ Row.RID_current;
16             ORID ⟵ Row.RID_past;
17             Update the value of RID_current with NRID;
18             Update the value of RID_past with CRID;
19             Add ORID to the unused RID pool;
20             Set RID_future = Ø;
21             Set RID-Flag = Ø;
22             Set Linked-TID = Ø;
23         else
24             return;
25     else
26         return;
```

On receiving the pseudo-IMSI, the AuC first retrieves the embedded TID, and searches its subscriber database for the corresponding IMSI. If the retrieved TID is not found (as might occur with a maliciously generated *location update* request), the AuC takes no further action; otherwise, the AuC retrieves the IMSI and compares the embedded TID with the value of TID_{future} for this IMSI. If they do not agree (including if the TID matches either TID_{past} or $TID_{current}$ for this IMSI), the AuC takes no further action; otherwise, the AuC performs the following steps. Algorithm 9.2 provides a pseudocode description of this process, in which '⟵' indicates an assignment that

is an atomic instruction and 'update' indicates a corresponding database update that varies depending on the implementation of the subscriber's database.

1. It deletes the value of TID_{past} for the retrieved IMSI, and adds the value of TID_{past} to the pool of unused TIDs.
2. It sets TID_{past} to equal $TID_{current}$, $TID_{current}$ to equal TID_{future}, and TID_{future} to null.
3. It checks the RID flag. If the flag is clear, the AuC takes no further action. Otherwise, it updates the RID information in its subscriber database as follows.
   (a) It deletes the value of RID_{past} for the retrieved IMSI, and adds this to the unused RID pool.
   (b) It sets RID_{past} to equal $RID_{current}$, and $RID_{current}$ to equal RID_{future}.
   (c) It sets RID_{future}, RID flag, and *Linked-TID* to null.

9.6.3.4 RID Flag Setting

The RID flag is used to indicate to the AuC that the RID should be updated at the next opportunity. The flag is cleared initially, and is also cleared during identity update (when the RID is changed); however, we do not specify when it is set, since this is a matter for network policy. A possible trigger for setting the flag would be if the network believes the RID may have been disclosed, e.g. if a *MAC-failure* token is received.

9.6.4 Modifications to USIM

To support the scheme, a USIM will need to store certain additional information. We propose that the pseudo-IMSI is stored in the existing file EF_{IMSI} (in fact this is necessary since the ME is assumed to be unaware of the scheme), and that the RID is stored in a new EF. We further suppose that the initial RID value is set during USIM personalisation.

We next describe how the new pseudo-IMSI and the novel error token are computed by the USIM.

9.6.4.1 New Identity Retrieval

On receipt of an authentication challenge, i.e. *RAND* and *AUTN*, the USIM proceeds using the standard AKA procedure. After successful completion of AKA, the USIM processes the *RAND* to retrieve the new identities as follows (see Figure 9.8(a)). Algorithm 9.3 provides a pseudocode description of this process.

Algorithm 9.3: USIM process in robust pseudo-IMSIs

Data: $RAND$, $AUTN$
Result: New pseudo-IMSI and RID, failure tokens

begin
  $K \longleftarrow$ *shared master key*;
  $SIMSI \longleftarrow$ *stored pseudo-IMSI*;
  $AMF_{new} \longleftarrow$ *dummy value set by specification*;
  $AK \longleftarrow f5(RAND, K)$;
  $SQN \longleftarrow AK \oplus AUTN.(SQN \oplus AK)$;
  $XMAC \longleftarrow f1(RAND, AUTN.AMF, SQN, K)$;
  $SRID \longleftarrow$ *stored RID*;
  if $XMAC \neq AUTN.MAC$ **then** `/* Generate novel error token */`
    $MAC\text{-}M \longleftarrow f1^{*}(SRID, SIMSI, AMF_{new}, K)$;
    $AUTM \longleftarrow SRID \parallel MAC\text{-}M$;
    return $AUTM$;
  if SQN *is out of range* **then**
    $SQN_{MS} \longleftarrow$ *Stored SQN*;
    $MAC\text{-}S \longleftarrow f1^{*}(AMF_{new}, SQN_{MS}, RAND, K)$;
    $AK_{new} \longleftarrow f5^{*}(RAND, K)$;
    $AUTS \longleftarrow (SQN_{MS} \oplus AK_{new}) \parallel MAC\text{-}S$;
    return $AUTS$;
  $ESQN_1 \longleftarrow SQN \parallel Pad_1$;
  $EK_1 \longleftarrow f5^{**}(ESQN_1, K)$;
  $Masked\text{-}TID \longleftarrow$ *appropriate part of RAND*;
  $Encoded\text{-}TID \longleftarrow Masked\text{-}TID \oplus EK_1$;
  $INSByte \longleftarrow$ *appropriate part of Encoded-TID*;
  $TID \longleftarrow$ *appropriate part of Encoded-TID*;
  $RIMSI \longleftarrow PLMN\text{-}ID \parallel TID$;
  if $RIMSI \neq SIMSI$ **then**
    Update stored pseudo-IMSI with $RIMSI$;
  if $INSByte = INS_{RID}$ **then** `/* Retrieve new RID */`
    $ESQN_2 \longleftarrow SQN \parallel Pad_2$;
    $EK_2 \longleftarrow f5^{***}(ESQN_2, K)$;
    $Masked\text{-}RID \longleftarrow$ *appropriate part of RAND*;
    $RRID \longleftarrow Masked\text{-}RID \oplus EK_2$;
    if $RRID \neq SRID$ **then**
      Update stored RID with $RRID$;
  if $INSByte = INS_{reset}$ **then** `/* Resynchronise pseudo-IMSI */`
    Reset the USIM;

1. It sets EK_1 to equal $f5^{**}(SQN \parallel Pad_1, K)$.

2. It retrieves *masked-TID* from $RAND$, and parses *masked-TID*$\oplus EK_1$ to obtain

the TID and instruction byte.

3. It compares the retrieved instruction byte with the three predefined values ($INS_{regular}$, INS_{RID} and INS_{reset}). If it equals any of these values then the USIM concatenates the received TID with the PLMN-ID to obtain the new pseudo-IMSI, and compares the concatenated value with the stored pseudo-IMSI (an essential step since the same TID may be received multiple times). If they are the same, the USIM takes no further action; otherwise, it keeps a record of the new pseudo-IMSI and later updates its pseudo-IMSI using the procedure described in Section 6.4.

   (a) If the received instruction byte equals INS_{RID}, then the USIM also performs the following steps.

      i. It sets EK_2 to equal $f5^{***}(SQN \parallel Pad_2, K)$. It retrieves *masked-RID* from $RAND$, and parses *masked-RID*$\oplus EK_2$ to obtain a RID.

      ii. It compares the retrieved RID with the stored RID (an essential step since the same RID may be received multiple times). If they are the same the USIM takes no further action; otherwise it updates its stored RID.

   (b) If the received instruction byte equals INS_{reset}, then the USIM immediately updates its pseudo-IMSI with the received pseudo-IMSI, using the procedure described in Section 6.4. This case will arise when an HN wishes to synchronise the pseudo-IMSI between the USIM and the HN, as described in Section 9.6.5.

9.6.4.2 Failure Token Generation

As mentioned in Section 9.6.2, we introduce a novel error token, the AUTM, to enable a USIM to report a *MAC-failure* arising during authentication. To construct $AUTM$, the USIM computes a 64-bit MAC, the *MAC-M*, as a function of its stored RID, the current pseudo-IMSI, the key K, and a dummy AMF, using the existing $f1^*$ function (see Figure 9.8(b)), and sets $AUTM$ to ($RID \parallel$ *MAC-M*) (see line 9–11 in Algorithm 9.3). The structure of $AUTM$ is similar to that of the *sync-failure* token $AUTS$, discussed in Section 3.4.2. When a USIM detects a *MAC-failure* in authentication, it computes $AUTM$, and reports it to the SN in the same way as an $AUTS$.

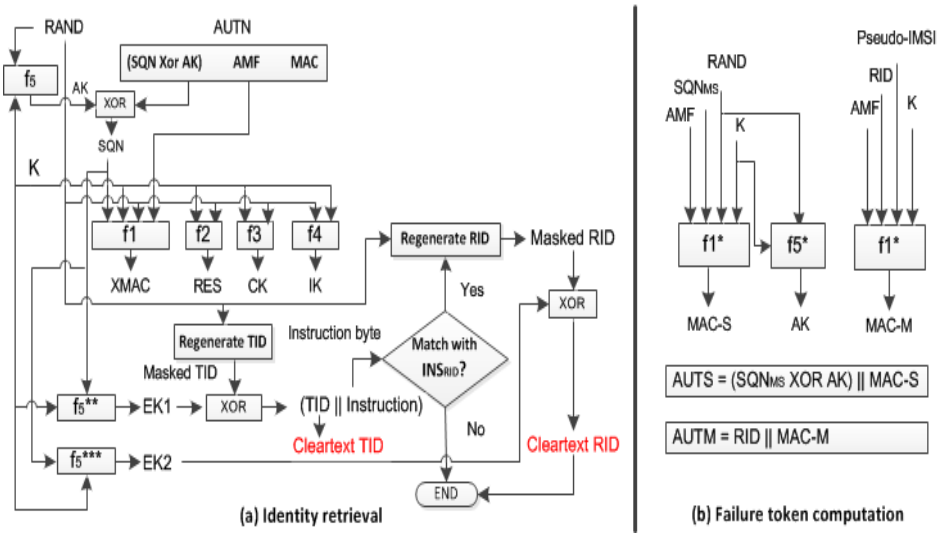

Figure 9.8: Computations in USIM for robust pseudo-IMSIs

9.6.5 Pseudo-IMSI Recovery

9.6.5.1 A Desynchronisation Scenario

Unfortunately, the robust pseudo-IMSIs scheme described here is not completely free from possible pseudo-IMSI desynchronisation. However, the scheme allows pseudo-IMSI synchronisation to be regained as soon as its loss is detected. Pseudo-IMSI desynchronisation can arises if the HN updates the subscriber's pseudo-IMSI but the corresponding USIM does not. Analogous to the discussion in Section 9.5.3, one possible scenario for such an event is as follows.

Suppose a malicious entity, e.g. a compromised SN, able to initiate a *location update* request sends such a request containing a randomly chosen pseudo-IMSI. If, by chance, the TID in this pseudo-IMSI happens to match a stored TID_{future}, then the HN will incorrectly update its subscriber database. If this 'new' TID has not been received by the corresponding USIM, then the current TID in the USIM will equal the value of TID_{past} in the HN database (see steps 1 and 2 of Section 9.6.3.3). Unlike the analogous scenario discussed in Section 9.5.3, this scenario does not cause pseudo-IMSI desynchronisation in the scheme described here, since the AuC is still able to map the IMSI to the TID currently stored in the USIM.

However, even in the scheme described here, the AuC will lose its mapping from the IMSI to the received pseudo-IMSI if the malicious entity could successfully cause another unauthorised TID update in the HN database. If pseudo-IMSI synchronisation is lost (and cannot be recovered), AKA will always fail, and the UE will not be able

to receive any network service until the USIM is replaced. The likelihood of such a pseudo-IMSI desynchronisation is reduced in the new scheme, in that it requires two false updates to the HN database; however, such an event is disastrous, and so a way to recover from this failure state is needed. We describe the recovery process in Section 9.6.5.2 below.

9.6.5.2 Synchronisation Recovery Process

In all the current mobile systems, there are scenarios in which various forms of desynchronisation can occur. For example, TMSI-IMSI synchronisation can be lost by an SN if it receives repeated *MAC-failure* messages as a result of failed authentications. In such a situation, the SN recovers by requesting the cleartext IMSI, obtaining a new set of AVs from the HN, running AKA, and then allocating a new TMSI.

Also, as discussed in Section 3.4.2, if a USIM identifies loss of SQN synchronisation between it and the HN, it resynchronises SQN by sending a *sync-failure* token to the HN. On receiving such a token, the HN adjusts its stored value of SQN, computes an AV with the new value, and sends the AV to the SN for use in AKA.

Analogously, in the scheme described here, if a USIM detects a possible pseudo-IMSI desynchronisation when authentication fails because of a MAC mismatch, it sends an $AUTM$ error token to the HN (via the SN). The SN sends other relevant information with the $AUTM$ or $AUTS$ token when sending it to the HN, notably the pseudo-IMSI (which the SN thinks is an IMSI) and the failure cause. As $AUTS$ and $AUTM$ are indistinguishable to the SN, the SN reports both types of token as a *sync-failure*.

On receiving the token, the AuC first runs the standard validation steps for an $AUTS$. If validation succeeds, the AuC performs the standard process for SQN recovery (as specified in Section 3.4.2); otherwise, it further verifies the token to check whether it is an $AUTM$. $AUTM$ checking involves the following steps.

1. It parses the token to retrieve a RID and *MAC-M*.
2. It searches its subscriber database for the IMSI for this RID. If the RID is not found, validation fails; otherwise, the AuC retrieves the IMSI and the corresponding K.
3. It computes MAC as $f1^*$*(RID, pseudo-IMSI, AMF, K)* using the pseudo-IMSI sent with the token by the SN, and compares it with the retrieved *MAC-M*. If they agree, the token is validated; otherwise, validation fails.

If $AUTM$ validation fails, the AuC reports the issue to the SN; otherwise, it rectifies the TID entries in its subscriber database using the procedure described in the following

paragraphs, computes an AV, and sends the AV to the SN for authentication.

As noted in the previous paragraph, if $AUTM$ validation succeeds, the SN must adjust its TID values appropriately; this is achieved as follows. The AuC first retrieves the TID from the pseudo-IMSI sent with the $AUTM$ token by the SN, and compares it with the three stored TID values (TID_{past}, $TID_{current}$ and TID_{future}) for this IMSI. If it equals any of the these values then the *MAC-failure* does not indicate that a pseudo-IMSI desynchronisation has occurred. Hence, in this case the AuC does not modify its subscriber database, computes an AV using the procedure described in Section 9.6.3.2, and sends the AV to the SN.

Otherwise, the AuC proceeds as follows (see also Algorithm 9.4).

1. It deletes the values of TID_{past} and $TID_{current}$ for the IMSI concerned, and adds them to the pool of unused TIDs.

2. It checks whether the TID retrieved from the SN-supplied pseudo-IMSI is in the pool of unused TIDs. If the TID is available, the AuC sets $TID_{current}$ to equal this TID, computes an AV using the procedure described in Section 9.6.3.2, and sends the AV to the SN. Otherwise, the pseudo-IMSI reported by the USIM must have been allocated to another subscriber, and the AuC performs the following steps (see steps 22 to 33 of Algorithm 9.4).

   (a) It computes EK_1 and selects a TID using steps 2 and 3 of Section 9.6.3.2.

   (b) The AuC sets $RAND$ to equal $((TID_{future} \parallel INS_{reset}) \oplus EK_1) \parallel randgen(80)$.

   (c) It generates an AV using the computed $RAND$ in the standard way, and sends the AV to the SN.

Note that a *MAC-failure* token reported by an honest subscriber as a result of a loss of synchronisation between its TMSI and the corresponding pseudo-IMSI will be deemed invalid by the HN. This is because the MAC component of the token is computed over the USIM's actual pseudo-IMSI, whereas the SN will send to the HN the (different) pseudo-IMSI it believes is currently associated with the TMSI used by the mobile device. To deal with this case, the SN could either use the HN's response in processing the failure token, or simply request the cleartext pseudo-IMSI before forwarding a failure token.

Algorithm 9.4: Pseudo-IMSI recovery in robust pseudo-IMSIs

Data: *Pseudo-IMSI, AUTM*
Result: An AV, synchronised pseudo-IMSI

begin
  $RRID \longleftarrow AUTM.RID$;
  Fetch relevant data (assume in Row structure) from subscriber database associated with $RRID$;
  $K \longleftarrow Row.K$;
  $IMSI \longleftarrow Row.IMSI$;
  $CTID \longleftarrow Row.TID_{current}$;
  $NTID \longleftarrow Row.TID_{future}$;
  $OTID \longleftarrow Row.TID_{past}$;
  $RTID \longleftarrow$ *TID part of the Pseudo-IMSI*;
  if $RTID \in \{OTID, CTID, NTID\}$ **then**
    Generate an AV using Algorithm 9.1;
    return AV;
  else
    Add OTID to the unused TID pool;
    Add CTID to the unused TID pool;
    Set $TID_{past} = \emptyset$;
    if $RTID$ *is unused* **then**
      Update the value of $TID_{current}$ *with* $RTID$;
      Generate an AV using Algorithm 9.1;
      return AV ;
    else `/* Generate a special AV */`
      Set $TID_{current} = \emptyset$;
      if $NTID \neq \emptyset$ **then** $TID_{fresh} \longleftarrow NTID$;
      else
        $TID_{fresh} \longleftarrow$ *an unused TID*;
        Update the value of TID_{future} *with* TID_{fresh};
      $SQN \longleftarrow$ *subscriber-specific SQN*;
      $ESQN_1 \longleftarrow SQN \parallel Pad_1$;
      $EK_1 \longleftarrow f5^{**}(ESQN_1, K)$;
      Encoded-TID $\longleftarrow TID_{fresh} \parallel INS_{reset}$;
      Masked-TID $\longleftarrow$ *Encoded-TID* $\oplus EK_1$;
      $RAND \longleftarrow$ *Masked-TID* $\parallel randgen(80)$;
      Generate AV using the $RAND$ *in the standard way*;
      return AV ;

9.7 Analyses of Robust Pseudo-IMSIs

9.7.1 Correctness of the Scheme

The scheme is an extension to the modifiable multiple IMSIs scheme described in Section 9.4; the security properties described in Section 9.5.1 therefore hold. We next

provide detailed arguments supporting this claim.

The modified AKA protocol is as secure as the existing AKA, since we have not modified the existing AKA except to replace the random *RAND* with a cryptographically constructed *RAND*. We formally verified this claim using ProVerif (see Section 9.8.2).

The constructed *RAND* has the same properties as a randomly (or pseudorandomly) chosen value. It is indistinguishable from a random value on the assumption that a data string masked using the output of the functions $f5^{**}$ and $f5^{***}$ is indistinguishable from a randomly chosen string, cf. [18, 19]. As for the modifiable multiple IMSIs scheme, the use of *SQN* randomises the pseudonym-encoded *RAND* if the same TID or RID is embedded in multiple *RAND* values.

For similar reasons as in the modifiable multiple IMSIs scheme, described in Section 9.5.1, cryptographic key generation is not affected.

The extended scheme provides the following security features:

1. pseudo-IMSIs and RIDs for the same subscriber are unlinkable;
2. a USIM will only accept a genuine pseudo-IMSI or a RID, i.e. a value originated by its HN; and
3. a request for pseudo-IMSI synchronisation recovery cannot be forged.

We next give more detailed arguments supporting these three claims.

9.7.1.1 Goal 1: Pseudo-IMSI/RID Unlinkability

The argument follows similar lines to that given in Section 9.5.1.1. Pseudo-IMSI and RID unlinkability depends on the following three assumptions.

1. The TID part of the pseudo-IMSI is randomly selected from an extensive list.
2. The RID is randomly selected from an extensive list;
3. The TID and the RID are confidentially communicated from the HN to the USIM.

Of these, assumptions 1 and 2 depend on correct implementation of the scheme, while assumption 3 depends on the confidentiality method that is used to protect a TID or a RID whilst in transit from an HN to a USIM. As described in Section 9.6.3.2, the TID and RID are protected in the same way as the MSIN part of a replacement IMSI is protected in the modifiable multiple IMSIs scheme. That is, both the TID and RID are masked by XORing them with a pseudorandom sequence output by a variant of

the $f5$ function. We formally verified that RID and TID confidentiality is maintained using ProVerif (see Section 9.8.2). Hence, an adversary without access to the key K is unable to learn a pseudo-IMSI or a RID before they are used, ensuring unlinkability between consecutive pseudo-IMSIs, and also between consecutive RIDs.

9.7.1.2 Goal 2: Pseudo-IMSI/RID Correctness

As for the modifiable multiple IMSIs scheme, new TIDs and RIDs are sent to a USIM using AKA, and a USIM will only accept a new RID or TID if AKA is successful, i.e. after the network (and hence the $RAND$) has been authenticated. Hence, AKA guarantees the origin, integrity and timeliness of the new pseudo-IMSI and RID.

Moreover, because the SQN value is checked by the USIM during AKA, an active adversary cannot force a USIM to accept an 'old' pseudo-IMSI or RID, since this checking forces AVs to be used in strict order of generation. That is, the timeliness of pseudo-IMSIs and RIDs is also guaranteed.

9.7.1.3 Goal 3: Pseudo-IMSI Synchronisation Correctness

This property holds because of the way in which a synchronisation recovery request is constructed and processed. As discussed in Section 9.6.5.2, the synchronisation recovery request contains an error token, the $AUTM$, computed by the USIM requesting synchronisation recovery. As described in Section 9.6.4.2, $AUTM$ contains a MAC computed over the USIM's current RID, pseudo-IMSI, and a value of AMF, using the key K. The MAC guarantees detection of malicious changes to the RID or pseudo-IMSI, preventing an adversary falsely initiating a pseudo-IMSI synchronisation recovery. We formally verified this claim using ProVerif (see Section 9.8.2). To do so we proved that if an HN initiates a pseudo-IMSI synchronisation recovery for a USIM, then the process must have been initiated by the appropriate USIM.

9.7.2 User Identity Confidentiality

The scheme diminishes the impact of IMSI catchers and improves user identity confidentiality by preventing the IMSI ever being sent across the air interface. However, air interface interactions are not completely anonymous, since the pseudo-IMSI functions as a pseudonym, potentially enabling the interactions of a single phone to be tracked for a period; of course, this is always true if a subscriber resides in a single location area, even when only a temporary identity, i.e. a TMSI or GUTI, is used.

Frequent AKA execution could lessen the impact of such tracking, which would also alleviate the problem of long TMSI validity periods over multiple geographic areas, as

reported by Arapinis et al. [47]. Although the requirement to execute AKA frequently is reduced in 4G, the importance of frequent AKA execution in preventing security attacks is discussed in recent research [101].

However, as described in Section 9.7.1.1, pseudo-IMSIs and RIDs for the same USIM are unlinkable, ensuring user identity confidentiality.

9.7.3 Identity Synchronisation

As discussed in Section 9.7.1.2, an adversary cannot force an unauthorised pseudo-IMSI change to occur. Nevertheless, an adversary can stop or delay the arrival of a *RAND* containing a new pseudo-IMSI at a USIM. However, such an event does not affect pseudo-IMSI desynchronisation, since, as in the modifiable multiple IMSIs scheme, the HN retransmits a pseudo-IMSI until the HN has reliable evidence that it has been received by the USIM.

Unlike the modifiable multiple IMSIs and van den Broek et al. schemes, described in Sections 9.4 and 9.5.6, an HN updates its subscriber database only when it receives a specific *location update* request from an SN, and it keeps the immediate past pseudo-IMSI for each subscriber. These changes help to minimise the likelihood of identity desynchronisation, as discussed in Section 9.6.5.1. Moreover, inclusion of a pseudo-IMSI synchronisation recovery process guarantees synchronisation of the pseudo-IMSI between a USIM and its HN.

Although RIDs and TIDs are managed in the same way, a RID is only used to recover pseudo-IMSI synchronisation. The frequency of RID updates is a policy matter for the network. It might be possible to deploy other methods to guarantee RID-IMSI synchronisation, a possible avenue for future research.

9.7.4 Synchronisation Recovery

As discussed in Section 9.7.1.3, the synchronisation recovery mechanism prevents an adversary falsely initiating a pseudo-IMSI recovery. We formally verified correctness of the synchronisation recovery process using ProVerif (see Section 9.8.2).

The pseudo-IMSI synchronisation recovery process is similar to the existing *SQN* synchronisation recovery process. Unlike for *SQN*, the RID in the pseudo-IMSI synchronisation recovery request is transferred in cleartext, since the HN is unable to identify the subscriber from the reported pseudo-IMSI while pseudo-IMSIs are desynchronised. This allows possible user tracking using the RID. As the RID changes over time, the threat of RID traceability is comparable to that arising from the current temporary identities (TMSI and GUTI).

9.7.5 Performance and Overhead

The scheme introduces minimal additional overhead to a USIM. We add two KDFs (to retrieve identities) and one MAC function (to support identity recovery), all of which are similar to the existing USIM functions. Transferring a new pseudo-IMSI to the ME is an additional task for a USIM, which could be performed when the ME is idle. This overhead seems likely to be manageable, even for a USIM with limited computational power.

The scheme requires the HN to manage new types of identifier. It increases database transactions, adds two KDFs for computing an AV, and introduces new functionalities, notably the need to refresh an identity on receiving an appropriate *location update* request and identity recovery in the event of pseudo-IMSI desynchronisation. Since none of these are particularly complex, it seems likely that these could be achieved with some combination of allocating more resources, clustering subscribers in multiple HSSs, and efficient database design.

As in the modifiable multiple IMSIs scheme, the scheme does not affect any functionality in the SN or introduce any additional communications. The only impact is an increase in the apparent number of subscribers at the SN, since subscribers switching to a new pseudo-IMSI appear like new subscribers.

Pseudo-IMSIs and IMSIs for a single HN must all be distinct. Since multiple pseudo-IMSIs are allocated for each subscriber, pressure could be created on the number of IMSIs available to an operator. To address this issue, the approaches described in Section 9.5.4 could be implemented.

9.7.6 Deployment and Interoperability

Like the modifiable multiple IMSIs scheme described in Section 9.4, the scheme modifies only the USIM and the HN, owned by a single entity, and is transparent to the SN and mobile phone. This allows phased deployment, e.g. by including the additional functionality in newly issued USIMs while existing USIMs continue to function as at present. In addition, the scheme does not affect existing services dependent on the IMSI, e.g. lawful interception and billing, making deployment simpler than for the modifiable multiple IMSIs scheme, as well as the van den Broek et al. scheme described in Section 9.5.6.

There are certain practical issues to be considered. For example, the set of 'normal' IMSIs used by existing USIMs needs to be kept distinct from the range of pseudo-IMSIs used by the new USIMs. Also, the HN should use location information from both the user's current and past pseudo-IMSI in supporting mobile terminated services; that is,

the HN might use the location information for the pseudo-IMSI containing TID_{past} if delivery of a mobile terminated service using the pseudo-IMSI containing $TID_{current}$ fails.

Like the modifiable multiple IMSIs scheme described in Section 9.4.2, the scheme will not work for GSM as it depends on the mutual authentication feature of 3G and 4G AKA. If a UE using a new-style USIM needs to connect to a GSM network, it should continue to use the fixed pseudo-IMSI as long as it is connected to that network. The pseudo-IMSI can be updated when the UE next roams to a 3G or 4G network.

9.7.7 Impact on Other Attacks

Arapinis et al. [48] describe a user linkability attack, discussed in Section 8.2.1, that allows an adversary to distinguish between UEs based on the error messages arising from a failed AKA execution. The change to the use of $AUTM$ for reporting *MAC-failures* renders them indistinguishable from *sync-failures*, hence invalidating this attack.

An attack on the USIM provision process could compromise the key K [145], thereby defeating all the security features. However, the scheme described here could reduce the impact of key compromise, since the initial mapping from a pseudo-IMSI to the key K is lost as soon as the subscriber changes its pseudo-IMSI, and even an adversary knowing K would not be able to readily track a device.

9.8 Formal Verification

In this section we describe details of formal verifications of the modifiable multiple IMSIs scheme described in Section 9.4 and the robust pseudo-IMSI scheme described in Section 9.6. We used the ProVerif tool introduced in Section 5.8.1 for these formal analyses. We modelled the modified AKA protocol for both schemes, and verified the security and privacy properties discussed in Sections 9.5 and 9.7.

9.8.1 Modifiable Multiple IMSIs

In this section we first describe a ProVerif model of the modifiable multiple IMSIs scheme and present a formalisation of the security and privacy properties of interest here. We then discuss the results obtained from the ProVerif analysis.

9.8.1.1 Formal Model

As described in Section 5.8.2, a ProVerif model of a protocol is divided into the declarations, process macros, and the main process; we start by providing a summary of

these parts of the model for the modifiable multiple IMSIs scheme — full details are provided in Appendix B.1. The ProVerif model we present immediately below is an extension of the model described in Section 5.8.2.

Listing 9.1 contains the salient parts of the declarations for the model of modifiable multiple IMSIs. The full listing is given in Appendix B.1. In Listing 9.1, the $f1$, $f2$, $f3$, $f4$, $f5$ constructors model the authentication-specific cryptographic functions used in AKA, and the $f11$ and $f51$ constructors model the additional cryptographic functions introduced in the scheme. The values *mac, nonce, key, resp, cipherKey, integrityKey, anonymityKey* and *maskKey* are user-defined types. The constructors *aencrypt* and *mencrypt* model the $\oplus$ operator used to conceal the *SQN* and *MSIN* values. As we used the same operation to conceal both the values, a single constructor could be used. However, we employed separate constructors for flexibility. To retrieve the *SQN* and *MSIN* value in the UE, the destructors *adecrypt* and *mdecrypt* are used.

Listing 9.1: Modifiable multiple IMSIs model: Summary of declarations

```
1  (* Constructors and destructors *)
2    fun f1(bitstring, mac, bitstring, nonce, key): mac.
3    fun f11(bitstring, key): mac.
4    fun f2(mac, bitstring, nonce, key): resp.
5    fun f3(mac, bitstring, nonce, key): cipherKey.
6    fun f4(mac, bitstring, nonce, key): integrityKey.
7    fun f5(mac, bitstring, nonce, key): anonymityKey.
8    fun f51(bitstring, key): maskKey.
9    fun aencrypt(bitstring, anonymityKey): bitstring.
10   fun mencrypt(bitstring, maskKey): bitstring.
11   reduc forall m: bitstring, k: anonymityKey; adecrypt(aencrypt(m, k
         ), k) = m.
12   reduc forall n: bitstring, l: maskKey; mdecrypt(mencrypt(n, l), l)
          = n.

14   free sqn: bitstring [private].
15   free msin: bitstring [private].

17 (* Secrecy queries *)
18   query attacker(sqn).
19   query attacker(msin).

21 (* Authentication queries *)
22   query x1: ident, x2: cipherKey, x3: integrityKey; event(endSN(x1,
         x2, x3)) ==> event(begSN(x1, x2, x3)).
23   query x1: ident, x2: cipherKey, x3: integrityKey; event(endUE(x1,
         x2, x3)) ==> event(begUE(x1, x2, x3)).
```

As described in Section 5.8.2.1, the declarations part also formalises the security properties to be verified. We are interested in verifying the secrecy property of *SQN* and of the transferred identity, i.e. an MSIN, when sent from the HN to the USIM. We are also interested in verifying that the scheme provides mutual authentication between the USIM and the SN. These are achieved by using the *reachability* and *correspondence assertion* queries, described below.

In lines 14 and 15 of Listing 9.1, we declared private *free* variables *sqn* and *msin*, representing the private data in the model. Lines 18, 19, 22, and 23 in Listing 9.1 query the secrecy (*reachability* query) and authenticity (*correspondence assertion* query) properties of interest in our analysis. The events *begSN*, *endSN*, *begUE*, *endUE* are described below, as part of the process description in which they are used.

As in the model described in Section 5.8.2, we define three process macros to model the processes of the UE, SN, and HN. The most significant parts of the UE process are given in Listing 9.2 below. Similarly, we assume that the communication channel between the UE and SN (the *pubChannel* variable in the model) is public and that the communication channel between the SN and HN (the *secureChannel* variable in the model) is private.

Listing 9.2: Modifiable multiple IMSIs model: UE process (highlights)

```
1  let processUE=
2   out(pubChannel, (ID, imsi_ms));
3   in(pubChannel, (=CHALLENGE, SMAC_ms: mac, masked_msin_ms: bitstring
       , r_ms: nonce, enc_sqn_ms: bitstring, mac_ms: mac));
4   let ak_ms: anonymityKey = f5(SMAC_ms, masked_msin_ms, r_ms, ki) in
5   let sqn_ms: bitstring = adecrypt(enc_sqn_ms, ak_ms) in
6   if f1(sqn_ms, SMAC_ms, masked_msin_ms, r_ms, ki) = mac_ms then
7    let res_ms: resp = f2(SMAC_ms, masked_msin_ms, r_ms, ki) in
8    let ck_ms: cipherKey = f3(SMAC_ms, masked_msin_ms, r_ms, ki) in
9    let ik_ms: integrityKey = f4(SMAC_ms, masked_msin_ms, r_ms, ki) in
10   event endUE(imsi_ms, ck_ms, ik_ms);
11   event begSN(imsi_ms, ck_ms, ik_ms);
12   out(pubChannel, (RES, res_ms)).
```

The UE process, specified in the *processUE* macro (see Listing 9.2), sends the subscriber identity across the public channel for authentication, and then waits for an authentication challenge (*RAND* and *AUTN*), in which *RAND* is the concatenation of the SMAC, the masked *MSIN* and a random number. On receiving the challenge, the process uses the destructor functions to retrieve *SQN*, to compute a MAC, and to compare it with the received MAC. If they agree, the SN has been successfully

authenticated; in such a case, the process marks the event *endUE* (line 10 in Listing 9.2) with the IMSI and the corresponding session keys as event parameters. The event *endUE* indicates that authentication of the SN by the UE has completed. The process also marks the event *begSN* (line 11 in Listing 9.2) with the IMSI and the corresponding session keys as event parameters and sends the computed *RES* across the public channel to be received by the SN process. The event *begSN* indicates that authentication of the UE by the SN has started.

SN protocol execution is described in the *processSN* macro (see Listing 9.3). On receiving the subscriber identity from the public channel, this process sends the identity across the private channel to the HN, and waits for an AV. When it receives the AV, it marks the event *begUE* (line 5 in Listing 9.3), with the IMSI and the corresponding session keys as event parameters, and sends the received authentication challenge across the public channel to the UE. Like the events described in *processUE*, the event *begUE* indicates that authentication of the SN by the UE has started. The process then waits for the *RES*, and on receiving a value from the public channel, it compares it with the *XRES* in the AV. If they agree then authentication of the UE to the SN is complete; in such a case, the process marks the event *endSN* (line 9 in Listing 9.3), with the IMSI and corresponding session keys as event parameters. The event *endSN* indicates that authentication of the UE by the SN has completed.

Listing 9.3: Modifiable multiple IMSIs model: SN process (highlights)

```
1  let processSN=
2    in(pubChannel, (=ID, imsi_sn: ident));
3    out(secureChannel, (AV_REQ, imsi_sn));
4    in(secureChannel, (=AV, imsi_hn_sn: ident, SMAC_sn: bitstring,
         masked_msin_sn: bitstring, r_sn: nonce, enc_sqn_sn: bitstring,
         mac_sn: mac, xres_sn: resp, ck_sn: cipherKey, ik_sn:
         integrityKey));
5    event begUE(imsi_hn_sn, ck_sn, ik_sn);
6    out(pubChannel, (CHALLENGE, SMAC_sn, masked_msin_sn, r_sn,
         enc_sqn_sn, mac_sn));
7    in(pubChannel, (=RES, res_sn: resp));
8    if res_sn = xres_sn then
9          event endSN(imsi_hn_sn, ck_sn, ik_sn).
```

HN protocol execution is modelled in the *processHN* macro (see Listing 9.4). The HN receives an AV request from the SN (line 2 in Listing 9.4), computes an AV using the defined constructors (lines 5–13 in Listing 9.4), and sends the AV to the SN (line 14 in Listing 9.4).

Listing 9.4: Modifiable multiple IMSIs model: HN process (highlights)

```
let processHN=
 in(secureChannel, (=AV_REQ, imsi_hn: ident));
 get keys(=imsi_hn, ki_hn) in
 new r_hn: nonce;
 let SMAC_hn: mac = f11(sqn, ki_hn) in
 let ek_hn: maskKey = f51(sqn, ki_hn) in
 let masked_msin_hn: bitstring = mencrypt(msin, ek_hn) in
 let mac_hn: mac = f1(sqn, SMAC_hn, masked_msin_hn, r_hn, ki_hn) in
 let xres_hn: resp = f2(SMAC_hn, masked_msin_hn, r_hn, ki_hn) in
 let ck_hn: cipherKey = f3(SMAC_hn, masked_msin_hn, r_hn, ki_hn) in
 let ik_hn: integrityKey = f4(SMAC_hn, masked_msin_hn, r_hn, ki_hn)
     in
 let ak_hn: anonymityKey = f5(SMAC_hn, masked_msin_hn, r_hn, ki_hn)
     in
 let enc_sqn_hn: bitstring = aencrypt(sqn, ak_hn) in
 out(secureChannel, (AV, imsi_hn, SMAC_hn, masked_msin_hn, r_hn,
     enc_sqn_hn, mac_hn, xres_hn, ck_hn, ik_hn)).
```

Just as in Section 5.8.2.3, the main process of the ProVerif model encodes the complete protocol; it captures the modified AKA protocol for the modifiable multiple IMSIs scheme using the macros defined above. Listing 9.5 shows the main process. A full listing of the ProVerif code for the model can be found in Appendix B.1.

Listing 9.5: Modifiable multiple IMSIs model: Main process

```
  process
        ((!processUE) | processSN | processHN)
```

9.8.1.2 Verification Results

We ran the encoded protocol in ProVerif to verify the secrecy and authenticity properties. The ProVerif tool successfully verified that the secrecy of *SQN* and *MSIN* is maintained, when transferred from the HN to the USIM. The tool output *RESULT not attacker(sqn[]) is true*, which means that the attacker is not able to learn the value of *SQN*. It also output *RESULT not attacker(msin[]) is true*, which means that the attacker is not able to learn the value of *MSIN*.

The ProVerif tool also verified mutual authentication between the USIM and the SN. In response to the first correspondence assertion query, the tool output *RESULT*

event(endSN$(x1_{1937}, x2_{1938}, x3_{1939})$) $\implies$ *event(begSN*$(x1_{1937}, x2_{1938}, x3_{1939})$) *is true*, which indicates that authentication of the UE by the SN is achieved. In response to the second correspondence assertion query, ProVerif returned *RESULT event(endUE(x1, x2, x3)) $\implies$ event(begUE(x1, x2, x3)) is true*, which indicates that network authentication by the UE is achieved. These two results imply that the mutual authentication property holds.

These proofs increase confidence that the scheme offers effective protection against IMSI catchers while maintaining the original functionality of the protocol. The ProVerif outputs for modifiable multiple IMSIs are given in Appendix B.3.1.

9.8.2 Robust Pseudo-IMSIs

In this section we first describe a ProVerif model of the robust pseudo-IMSIs scheme and present a formalisation of the security and privacy properties of interest here. We then discuss the results obtained from the ProVerif analysis.

9.8.2.1 Formal Model

We extend the model described in Section 9.8.1.1. The objective of the modelling process is to verify that the following security and privacy properties hold:

- the *SQN* remains confidential;
- the TID and the RID remain confidential when transferred from the HN to the USIM;
- mutual authentication between the USIM and the SN is provided;
- correctness of the synchronisation recovery process is maintained.

To establish the desired properties we follow the same approach as in Section 9.8.1.1, i.e. we use *reachability* and *correspondence assertion* queries. We next provide a summary of the model — full details are provided in Appendix B.2.

Listing 9.6 contains the most significant parts of the declarations part of the model of the robust pseudo-IMSIs scheme. The full listing is given in Appendix B.2. In Listing 9.6, the $f1$, $f2$, $f3$, $f4$, $f5$ constructors model the authentication-specific cryptographic functions used in AKA. The constructor $f12$ models the MAC computation required to construct an error token. The constructors $f52$ and $f53$ model the generation of the two types of additional masking keys. The values *nonce, key, mac, ident, resp, cipherKey, integrityKey, anonymityKey* and *maskKey* are user-defined types. The constructors *aencrypt* and *mencrypt* model the $\oplus$ operator used to conceal the

Listing 9.6: Robust pseudo-IMSIs model: Summary of declarations

```
1  (* Constructors and destructors *)
2   fun f1(bitstring, bitstring, bitstring, nonce, key): mac.
3   fun f12(bitstring, ident, key): mac.
4   fun f2(bitstring, bitstring, nonce, key): resp.
5   fun f3(bitstring, bitstring, nonce, key): cipherKey.
6   fun f4(bitstring, bitstring, nonce, key): integrityKey.
7   fun f5(bitstring, bitstring, nonce, key): anonymityKey.
8   fun f52(bitstring, key): maskKey.
9   fun f53(bitstring, key): maskKey.
10  fun aencrypt(bitstring, anonymityKey): bitstring.
11  fun mencrypt(bitstring, maskKey): bitstring.
12  reduc forall m: bitstring, k: anonymityKey; adecrypt(aencrypt(m, k)
        , k) = m.
13  reduc forall n: bitstring, l: maskKey; mdecrypt(mencrypt(n, l), l)
       = n.

15  free sqn: bitstring [private].
16  free tid: bitstring [private].

18 (* Queries *)
19  query attacker(sqn).
20  query attacker(tid).
21  query x1: ident, x2: cipherKey, x3: integrityKey; event(endSN(x1,
       x2, x3)) ==> event(begSN(x1, x2, x3)).
22  query x1: ident, x2: cipherKey, x3: integrityKey; event(endUE(x1,
       x2, x3)) ==> event(begUE(x1, x2, x3)).
23  query x1: ident, x2: bitstring; event(endRecoveryHN(x1, x2)) ==>
       event(begRecoveryHN(x1, x2)).
```

SQN, TID and RID values. To retrieve the *SQN*, TID and RID values at the UE, the destructors *adecrypt* and *mdecrypt* are used.

In lines 15 and 16 of Listing 9.6, private *free* variables *sqn* and *tid* are declared. Lines 19 and 20 in Listing 9.6 query the secrecy property for the *SQN* and TID. We did not include a query for the secrecy property of RID, since the synchronisation recovery process is included in the model, in which a RID is transmitted in cleartext across the air interface; that is, the property cannot be established. However, since the RID is transferred to the USIM in exactly the same way as TID, the fact that we can establish the secrecy of the TID during its transfer, means that the same property holds for the RID during transfer if not when it is used.

Lines 21 and 22 in Listing 9.6 query the correspondence of events to establish the success of mutual authentication between the USIM and the SN. Similarly, line 23

queries the correspondence of events to prove the correctness of the synchronisation recovery process. The events *begSN*, *endSN*, *begUE*, *endUE*, *begRecoveryHN*, and *endRecoveryHN* are described below in the specifications of the processes in which they are used.

In the model, we assume that communications between the UE and SN are public and that communications between the SN and HN are private. The most significant parts of the UE process are given in Listing 9.7 below. The UE process, described in *processUE* macro (see Listing 9.7), sends the subscriber identity across the public channel for authentication, and then waits for an authentication challenge (*RAND* and *AUTN*), in which *RAND* is the concatenation of the masked RID, the masked TID and a random number.

Listing 9.7: Robust pseudo-IMSIs model: UE process (highlights)

```
1  let processUE=
2   out(pubChannel, (ID, imsi_ms));
3   in(pubChannel, (=CHALLENGE, masked_rid_ms: bitstring, masked_tid_ms
        : bitstring, r_ms: nonce, enc_sqn_ms: bitstring, mac_ms: mac));
4   new d_mac: mac;
5   new d_rid: bitstring;
6   let ak_ms: anonymityKey = f5(masked_rid_ms, masked_tid_ms, r_ms, ki
        ) in
7   let sqn_ms: bitstring = adecrypt(enc_sqn_ms, ak_ms) in
8   if f1(sqn_ms, masked_rid_ms, masked_tid_ms, r_ms, ki) = mac_ms then
         (
9    let res_ms: resp = f2(masked_rid_ms, masked_tid_ms, r_ms, ki) in
10   let success = true in
11         event endUE(imsi_ms, ck_ms, ik_ms);
12         event begSN(imsi_ms, ck_ms, ik_ms);
13   out(pubChannel, (RES, success, res_ms, d_rid, d_mac)))
14  else (
15   new d_res: resp;
16   let success = false in
17   let ek2_ms: maskKey = f53(sqn_ms, ki) in
18   let rid_ms: bitstring = mdecrypt(masked_rid_ms, ek2_ms) in
19   let mac_m_ms: mac = f12(rid_ms, imsi_ms, ki) in
20     event begRecoveryHN(imsi_ms, rid_ms);
21   out(pubChannel, (RES, success, d_res, rid_ms, mac_m_ms))).
```

On receiving the challenge, the UE process uses the destructor functions to retrieve *SQN*, to compute a MAC, and to compare it with the received MAC. If they agree, the SN has successfully been authenticated; in such a case, the process computes *RES* (line 9 in Listing 9.7) and marks the event *endUE* (line 11 in Listing 9.7) with the IMSI

and the corresponding session keys as event parameters. The event *endUE* indicates that authentication of the SN by the UE has completed. The process also marks the event *begSN* (line 12 in Listing 9.7) with the IMSI and the corresponding session keys as event parameters and sends the computed *RES* across the public channel to be received by the SN process. The event *begSN* indicates that authentication of the UE by the SN has started.

Otherwise, i.e. if the MACs do not agree, then authentication of the SN by the UE has failed, and the UE process computes an error token (lines 17–19 in Listing 9.7). The process also marks the event *begRecoveryHN* (line 20 in Listing 9.7) with the IMSI and the corresponding RID as event parameters and sends the error token across the public channel to be received by the SN process. The event *begRecoveryHN* indicates that synchronisation recovery has started in the UE.

SN protocol execution is specified in the *processSN* macro (see Listing 9.8). On receiving the subscriber identity from the public channel, the SN process sends the identity across the private channel to the HN, and waits for an AV. When it receives the AV, it marks the event *begUE* (line 5 in Listing 9.8), with the IMSI and the corresponding session keys as event parameters, and sends the authentication challenge from the received AV across the public channel to the UE. The event *begUE* indicates that authentication of the SN by the UE has started. The process then waits for an authentication response from the UE. On receiving a value from the public channel, it first verifies whether the received value is a *RES* or an error token. If the received value is a *RES*, the process compares it with the *XRES* in the AV. If they agree then authentication of the UE is completed; in such a case, the process marks the event *endSN* (line 9 in Listing 9.8), with the IMSI and corresponding session keys as event parameters. The event *endSN* indicates that authentication of the UE by the SN has completed. Otherwise, the received value is an error token, and the SN process forwards the value to the HN.

HN protocol execution is specified in the *processHN* macro (see Listing 9.9). The HN receives an AV request from the SN (line 2 in Listing 9.9), computes an AV using the defined constructors (lines 7–16 in Listing 9.9), and sends the AV to the SN (line 17 in Listing 9.9). The HN process then waits for an error token from the SN.

On receiving the token (line 19 in Listing 9.9), the HN process verifies it (lines 20–23 in Listing 9.9). If the verification is successful, the HN process marks the event *endRecoveryHN* (line 24 in Listing 9.9) with the IMSI and the corresponding RID as event parameters. The event *endRecoveryHN* indicates that synchronisation recovery has completed in the HN.

Listing 9.8: Robust pseudo-IMSIs model: SN process (highlights)

```
let processSN=
 in(pubChannel, (=ID, imsi_sn: ident));
 out(secureChannel, (AV_REQ, imsi_sn));
 in(secureChannel, (=AV, imsi_hn_sn: ident, masked_rid_sn: bitstring
     , masked_tid_sn: bitstring, r_sn: nonce, enc_sqn_sn: bitstring,
     mac_sn: mac, xres_sn: resp, ck_sn: cipherKey, ik_sn:
     integrityKey));
        event begUE(imsi_hn_sn, ck_sn, ik_sn);
 out(pubChannel, (CHALLENGE, masked_rid_sn, masked_tid_sn, r_sn,
     enc_sqn_sn, mac_sn));
 in(pubChannel, (=RES, success_sn: bool, res_sn: resp, rid_sn:
     bitstring, mac_m_sn: mac));
 if success_sn = true then
        ( if res_sn = xres_sn then event endSN(imsi_hn_sn, ck_sn,
            ik_sn))
 else
        out(secureChannel, (ERROR, imsi_hn_sn, rid_sn, mac_m_sn)).
```

Just as in the model described in Section 9.8.1.1, we model the modified AKA protocol using the macros defined above. Listing 9.10 gives the main process. A full listing of the ProVerif code for the model can be found in Appendix B.2.

9.8.2.2 Verification Results

We ran the protocol model in ProVerif to analyse the security and privacy properties of interest. The ProVerif tool successfully verified that the secrecy of *SQN* and TID is maintained, when transferred from the HN to the USIM. The tool output *RESULT not attacker(sqn[]) is true*, which means that the attacker is not able to learn the value of *SQN*. Similarly, it output *RESULT not attacker(tid[]) is true*, which means that the attacker is not able to learn the value of TID when it is transferred from the HN to the USIM. As discussed in Section 9.8.2.1, the model does not include a means to verify that RID secrecy is preserved; as discussed there, this is because the RID is revealed if it is used for identity synchronisation recovery (i.e. its secrecy cannot be guaranteed if it is ever used). However, also as discussed in Section 9.8.2.1, the fact that we can establish the secrecy of the TID during its transfer, means that the same property holds for the RID during transfer.

The ProVerif tool verified that mutual authentication is achieved between the USIM and the SN. In response to the first correspondence assertion query, the tool output *RESULT event(endSN*$(x1_{2891}, x2_{2892}, x3_{2893})$ $\implies$ *event(begSN*$(x1_{2891}, x2_{2892}, x3_{2893})$

Listing 9.9: Robust pseudo-IMSIs model: HN process (highlights)

```
let processHN=
 in(secureChannel, (=AV_REQ, imsi_hn: ident));
 get keys(=imsi_hn, ki_hn) in
 new r_hn: nonce;
 new rid_hn: bitstring;
 insert rids(rid_hn, imsi_hn);
 let ek2_hn: maskKey = f53(sqn, ki_hn) in
 let masked_rid_hn: bitstring = mencrypt(rid_hn, ek2_hn) in
 let ek_hn: maskKey = f52(sqn, ki_hn) in
 let masked_tid_hn: bitstring = mencrypt(tid, ek_hn) in
 let mac_hn: mac = f1(sqn, masked_rid_hn, masked_tid_hn, r_hn, ki_hn
     ) in
 let xres_hn: resp = f2(masked_rid_hn, masked_tid_hn, r_hn, ki_hn)
     in
 let ck_hn: cipherKey = f3(masked_rid_hn, masked_tid_hn, r_hn, ki_hn
     ) in
 let ik_hn: integrityKey = f4(masked_rid_hn, masked_tid_hn, r_hn,
     ki_hn) in
 let ak_hn: anonymityKey = f5(masked_rid_hn, masked_tid_hn, r_hn,
     ki_hn) in
 let enc_sqn_hn: bitstring = aencrypt(sqn, ak_hn) in
 out(secureChannel, (AV, imsi_hn, masked_rid_hn, masked_tid_hn, r_hn
     , enc_sqn_hn, mac_hn, xres_hn, ck_hn, ik_hn));

 in(secureChannel, (=ERROR, imsi_hn_s: ident, r_rid_hn: bitstring,
     mac_m_hn: mac));
 get rids(=r_rid_hn, imsi_hn2) in
 if imsi_hn_s = imsi_hn2 then
        get keys(=imsi_hn2, ki_hn2) in
        if f12(r_rid_hn, imsi_hn2, ki_hn2) = mac_m_hn then
                event endRecoveryHN(imsi_hn2, r_rid_hn).
```

Listing 9.10: Robust pseudo-IMSIs model: Main process

```
  process
        ((!processUE) | processSN | processHN)
```

is true, which indicates that authentication of the UE by the SN is achieved. In response to the second correspondence assertion query, ProVerif returned *RESULT event(endUE(x1, x2, x3)) $\implies$ event(begUE(x1, x2, x3)) is true*, which implies that network authentication by the UE is achieved.

ProVerif also verified correctness of the synchronisation recovery process; this is

achieved by the response to the third correspondence assertion query described in Section 9.8.2.1 (see line 23 in Listing 9.6). The tool output *RESULT event(endRecoveryHN* $(x1_{5235}, x2_{5236}))$ $\implies$ *event(begRecoveryHN*$(x1_{5235}, x2_{5236}))$ *is true*, which indicates that if the HN initiates a synchronisation recovery for a USIM, it must have been requested by the legitimate USIM.

These proofs increase confidence that the scheme offers effective protection against IMSI catchers while maintaining the original functionality of the protocol. The proof of correctness for pseudo-IMSI synchronisation recovery gives assurance that the scheme is able to regain pseudo-IMSI synchronisation without introducing any new vulnerabilities. The ProVerif outputs for robust pseudo-IMSIs are given in Appendix B.3.2.

9.9 Relationship to the Prior Art

The schemes described in this chapter are designed with the objective of enhancing user privacy in current mobile systems, whilst avoiding significant modifications to the already deployed infrastructure. While other authors have observed that significant changes to widely deployed infrastructure are unlikely to be feasible, very few realistic and practical proposals for privacy enhancements which do not require such changes have been made (as discussed in Section 7.4).

Indeed, apart from the recent work of van den Broek, Verdult and de Ruiter (described in Section 9.5.6) and the work of Barbeau and Robert (described in Section 7.4.2), we know of no other proposed modifications to the operation of 3G and 4G with the objective of enhancing user privacy that do not involve major changes to the network infrastructure. However, both of these schemes are susceptible to possible identity desynchronisation, which is a critical threat to the operation of the 3G and 4G mobile systems.

To the best of the author's knowledge, the robust pseudo-IMSIs scheme described in this chapter is the first proposed scheme using a pseudonymous IMSI that both incorporates IMSI synchronisation recovery and minimises modifications to the deployed infrastructure. However, as discussed in Section 7.4.2, other pseudonym-based schemes designed to enhance user privacy have been proposed, although they all require changes to all the main components of the system, including HNs, SNs, phones and USIMs.

9.10 Summary

In this chapter we have described two approaches to using multiple IMSIs for a 3G/4G USIM. The goal of both schemes is to improve user privacy by reducing the impact

of IMSI disclosure on the air interface. They do not require any changes to the existing deployed network infrastructures, i.e. to the SN, air interface protocols or mobile devices. The overhead introduced is modest and should be feasible to manage in real-world networks. One major advantage is that the proposed schemes could be deployed immediately since they are completely transparent to the existing deployed infrastructure.

We also described a novel and robust authentication scheme for 3G and 4G that does not affect existing SNs and mobile phones, and preserves IMSI confidentiality. That is, the IMSI is never sent across a communication channel; instead a changing pseudo-IMSI is used. The pseudo-IMSIs appear as new subscribers to the SN, and are unlinkable. The scheme also addresses the possible loss of pseudo-IMSI synchronisation between USIM and HN, by incorporating an approach for pseudo-IMSI synchronisation recovery to be used if pseudo-IMSI synchronisation is lost.

Both of the schemes introduce changes to the operation of the HN and USIM, but not to the SN, the mobile device, or other internal network protocols, which enables transparent migration. We discussed the strengths and limitations of the schemes, and reported the results of formal analyses using ProVerif. The schemes alleviate the decades-old privacy problem of IMSI disclosure on the air interface, and hence 'trash the IMSI catchers'.

Part IV

Conclusion

Overview

Part IV concludes the thesis by summarising the main contributions as well as highlighting possible areas for future work. This part of the thesis consists of a single chapter, *Chapter 10*.

Chapter 10

Conclusions and Possible Future Work

In this chapter we summarise both the research achievements and possible limitations of the work described in this thesis. We also discuss opportunities for future work.

10.1 Conclusions

The overall goal of this thesis is to propose techniques for improving the security and privacy properties of current mobile systems without changing the deployed network infrastructure, i.e. the serving networks and mobile phones. The first step was to critically review prior research on the security and privacy of 2G, 3G and 4G mobile systems. As has been widely discussed, there are significant security and privacy threats arising from use of such systems. This motivated the work described in this thesis designed to address the security threats stemming from lack of mutual authentication in 2G GSM and the privacy threat arising from IMSI disclosure that applies to all mobile systems.

In Chapters 2 and 3, we described those aspects of the system architecture, user identities, air interface protocols, and services of 2G, 3G, and 4G systems necessary to understand the work described in this thesis.

In Chapter 4, we described the security and privacy threats arising from use of GSM. We critically analysed previously proposed schemes to address these threats. We did not find any previously proposed scheme that adds mutual authentication to GSM that does not require significant changes to the operation of all the entities in a GSM network. Similarly, almost all the previously proposed system enhancements designed to enhance user identity privacy protection in GSM require major changes

to the operation of all networks, including modifications to the serving networks and mobile equipment (i.e. mobile phones). Since any changes to the operation of the serving network have a very major impact on global GSM networks, it seems likely that such changes after deployment are infeasible in practice. The only scheme designed to enhance user identity privacy protection in GSM that does not require any changes to the existing deployed network infrastructures is the scheme, developed in parallel to the modifiable multiple IMSIs scheme described in this thesis, proposed by van den Broek, Verdult and de Ruiter (described in Section 6.6). However, as discussed in Section 9.5.7, this scheme appears to permit loss of identity synchronisation between the SIM and the home network, which is likely to lead to a permanent loss of service. As a result, it is unlikely that any of the schemes will ever be deployed in practice.

In Chapter 5, we proposed an enhanced GSM AKA scheme to provide authentication of the network to the mobile station, complementing the MS-to-network authentication already provided. This provides protection against some of the most serious threats to the security of GSM networks. This is achieved in a way which leaves the existing serving network infrastructure unchanged, and also does not require any changes to existing mobile phones. That is, unlike previously proposed schemes, it is practically realisable. The scheme only requires SIMs and the home network to be upgraded. Since both the SIM and the home network are managed by a single entity in mobile systems, such a solution can be rolled out piecemeal with no impact on the existing global infrastructure. We also analysed the proposed modification to GSM AKA using the ProVerif tool. The analysis confirmed that the modified protocol provides mutual authentication without leaking any confidential data.

One limitation of the proposed scheme is the requirement that mobile devices must support the 'class e' STK commands. Other possible limitations are that AVs must be used in the correct order, and that it is not clear what happens if the serving network never performs AKA. These issues therefore need further investigation, as discussed in Section 10.2 below.

In Chapter 6, we introduced two general approaches (i.e. the predefined multiple IMSIs scheme and the modifiable multiple IMSIs scheme) to using multiple IMSIs for a mobile subscriber intended to improve air interface user privacy. We described a privacy-enhancing scheme for GSM which does not require modifications to the existing deployed network infrastructures, i.e. to the SN, air interface protocols or mobile devices. We reported the results of the experiments to validate part of the schemes.

In Chapter 7, we analysed a wide range of research addressing the problem of the lack of robust user identity privacy in 3G and 4G. Unfortunately, just as for GSM, almost all the previously proposed schemes require changes to all the main components

of the system, including home networks, serving networks, phones and USIMs. This essentially means that deploying them would require implementing an entirely new network infrastructure, which seems unlikely to occur. The two schemes which do not require such a major redeployment are the second scheme of Barbeau and Robert (described in Section 7.4.2) and a recent proposal of van den Broek, Verdult and de Ruiter (described in Section 9.5.6). However, as we discussed, both the schemes appear to permit loss of identity synchronisation between the USIM and the home network, which is likely to lead to a permanent loss of service. As a result, these solutions too are unlikely to be deployable in practice.

In Chapter 8, we critically analysed the proposals of Arapinis et al. [48], intended to address a range of user privacy threats. This analysis revealed that the proposed modifications are impractical in a variety of ways; not only are there security and implementation issues, but the necessary changes to the operation of the system are very significant and much greater than was envisaged by the authors. In fact, some of the privacy issues appear almost impossible to address without a complete redesign of the security system, meaning that making significant system changes to address some of them are unlikely to be worth the effort. The shortcomings of the proposed 'fixes' exist despite the fact that the modifications have been verified using a logic-based modelling tool, suggesting that such tools need to be used with great care. We also suggested possible alternative approaches to some of the modifications.

In Chapter 9, we described two approaches (i.e. the predefined multiple IMSIs scheme and the modifiable multiple IMSIs scheme) to using multiple IMSIs for a mobile subscriber in 3G and 4G. The goal of these proposals is to improve user privacy by reducing the impact of IMSI disclosure on the air interface. These approaches do not require any changes to the existing deployed network infrastructures, i.e. to the serving networks, air interface protocols or mobile devices. The only changes required are to the operation of the authentication centre in the home network and to the USIM, both owned by a single entity. The overhead introduced is modest and should be feasible to manage in real-world networks. One major advantage is that the proposed schemes can be deployed immediately since they are completely transparent to the existing deployed infrastructure.

We also described a related scheme, developed in parallel to the modifiable multiple IMSIs scheme described in Section 9.4, proposed by van den Broek, Verdult and de Ruiter (see Section 9.5.6). Both the modifiable multiple IMSIs and van den Broek et al. schemes have limitations; in particular, both schemes appear to permit possible loss of identity synchronisation between the USIM and the home network, which is likely to lead to a permanent loss of service.

We enhanced the modifiable multiple IMSIs scheme to address the possible shortcomings. This enhanced scheme incorporates two major improvements: it addresses possible loss of identity synchronisation, and avoids the need for an HN to change its subscriber's IMSI by using instead a pseudo-IMSI (which is indistinguishable from an IMSI to a serving network). We present an approach for identity recovery to be used in the event of pseudo-IMSI desynchronisation. The scheme requires changes to the home network and the USIM, but not to the serving network, mobile phone or other internal network protocols, enabling simple, transparent and evolutionary migration. We discussed the strengths and limitations of both schemes, and verified their correctness and security properties using ProVerif.

One limitation of the pseudonymous IMSIs schemes is the pressure it potentially causes on the number of IMSIs available to a network operator. The schemes also involve additional overhead for the operation of the home network. These issues merit further examination, as discussed in Section 10.2.

To conclude, the work described in this thesis is an important contribution in improving the security and privacy features in current mobile systems. We strongly believe that significant changes to widely deployed infrastructures are unlikely to be feasible; therefore, a realistic and practical proposal for improving security and privacy in current mobile systems must do so in such a way that major infrastructure changes are not required, and this is what we have achieved.

10.2 Future Work

We conclude the thesis by highlighting possible areas for future work.

- There are a number of aspects of the GSM mutual authentication scheme described in Chapter 5 that require further investigation. Of particular significance is the behaviour of mobile devices in networks that do not perform the AKA protocol. Understanding mobile device behaviour in such circumstances is of importance in all mobile networks, not just GSM.

- In the robust pseudo-IMSIs scheme, the RID is solely used in pseudo-IMSI synchronisation recovery. The RID and TID are managed in the same way. However, it might be possible to deploy other methods to guarantee synchronisation between RID and IMSI, a possible avenue for future research.

- Simulating the proposed protocols in test networks with the support of real operators would increase confidence in their feasibility. It would also enable the overhead on the home network to be better quantified.

- It would be of interest to investigate the business and technical aspects of deciding when, and how often, to trigger identity updates, notably the changeable IMSI in the modifiable multiple IMSIs scheme, and the RID in the robust pseudo-IMSIs scheme.

- We have (implicitly) assumed that the protocols running in the core network are secure; this seems reasonable given that the core network is only accessible to trusted parties. However, in the future with likely changes in modes and business models for service delivery, this assumption may no longer hold. It would therefore be of interest to investigate what threats arise and what mitigations might be deployed, if assumptions about the integrity and security of the core network are weakened.

Bibliography

[1] 3rd Generation Partnership Project (3GPP). *3GPP TR 33.902 V4.0.0 : Technical Report; Technical Specification Group Services and System Aspects; 3G Security; Formal Analysis of the 3G Authentication Protocol (Release 4)*, September 2001.

[2] 3rd Generation Partnership Project (3GPP). *3GPP TS 03.20 V9.0.0 : Technical Specification; Digital cellular telecommunications system (Phase 2+); Security related network functions (GSM 03.20 Release 2000)*, January 2001.

[3] 3rd Generation Partnership Project (3GPP). *3GPP TS 21.133 V4.1.0 : Technical Specification; Technical Specification Group Services and System Aspects; 3G Security; Security Threats and Requirements (Release 4)*, December 2001.

[4] 3rd Generation Partnership Project (3GPP). *3GPP TS 42.017 V4.0.0 : Technical Specification; Technical Specification Group Terminals; Subscriber Identity Modules (SIM); Functional characteristics (Release 4)*, March 2001.

[5] 3rd Generation Partnership Project (3GPP). *3GPP TS 03.03 V7.8.0 : Technical Specification; Technical Specification Group Core Network; Numbering, addressing and identification; (GSM 03.03 Release 1998)*, September 2003.

[6] 3rd Generation Partnership Project (3GPP). *3GPP TS 03.08 V7.5.0 : Technical Specification; Technical Specification Group Core Network; Digital cellular telecommunications system (Phase 2+); Organisation of subscriber data (GSM 03.08 Release 1998)*, June 2003.

[7] 3rd Generation Partnership Project (3GPP). *3GPP TS 04.08 V7.21.0 : Technical Specification; Technical Specification Group Core Network; Mobile radio interface layer 3 specification (GSM 04.08 Release 1998)*, December 2003.

[8] 3rd Generation Partnership Project (3GPP). *3GPP TS 09.02 V7.15.0 : Technical Specification; Technical Specification Group Core Network; Mobile Application Part (MAP) specification (GSM 09.02 Release 1998)*, March 2004.

[9] 3rd Generation Partnership Project (3GPP). *3GPP TS 51.014 V4.5.0 : Technical Specification; Specification of the SIM Application Toolkit for the Subscriber Identity Module–Mobile Equipment (SIM–ME) interface (Release 4)*, December 2004.

[10] 3rd Generation Partnership Project (3GPP). *3GPP TS 51.011 V4.15.0 : Technical Specification; Technical Specification Group Core Network and Terminals; Specification of the Subscriber Identity Module–Mobile Equipment (SIM–ME) interface (Release 4)*, June 2005.

[11] 3rd Generation Partnership Project (3GPP). *3GPP TS 02.09 V8.1.0 : Technical Specification; Technical Specification Group Services and System Aspects; Security aspects (GSM 02.09 Release 1999)*, June 2006.

[12] 3rd Generation Partnership Project (3GPP). *3GPP TS 11.11 V8.14.0 : Technical Specification; Technical Specification Group Terminals Specification of the Subscriber Identity Module–Mobile Equipment (SIM–ME) interface (GSM 11.11 Release 1999)*, June 2007.

[13] 3rd Generation Partnership Project (3GPP). *3GPP TS 11.14 V8.18.0 : Technical Specification; Specification of the SIM Application Toolkit for the Subscriber Identity Module–Mobile Equipment (SIM–ME) interface (GSM 11.14 Release 1999)*, June 2007.

[14] 3rd Generation Partnership Project (3GPP). *3GPP TR 33.821 V9.0.0 : Technical Report; Technical Specification Group Services and System Aspects; Rationale and track of security decisions in Long Term Evolved (LTE) RAN / 3GPP System Architecture Evolution (SAE) (Release 9)*, June 2009.

[15] 3rd Generation Partnership Project (3GPP). *3GPP TS 23.012 V13.0.0 : Technical Specification; Technical Specification Group Core Network and Terminals; Location management procedures (Release 13)*, December 2015.

[16] 3rd Generation Partnership Project (3GPP). *3GPP TS 23.101 V13.0.0 : Technical Specification; Technical Specification Group Services and System Aspects; General Universal Mobile Telecommunications System (UMTS) architecture (Release 13)*, December 2015.

[17] 3rd Generation Partnership Project (3GPP). *3GPP TR 31.900 V13.0.0 : Technical Report; Technical Specification Group Core Network and Terminals; SIM/USIM internal and external interworking aspects (Release 13)*, January 2016.

[18] 3rd Generation Partnership Project (3GPP). *3GPP TR 35.909 V13.0.0 : Technical Report; Technical Specification Group Services and System Aspects; 3G Security; Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 5: Summary and results of design and evaluation (Release 13)*, January 2016.

[19] 3rd Generation Partnership Project (3GPP). *3GPP TR 35.934 V13.0.0 : Technical Specification; Technical Specification Group Services and System Aspects; Specification of the TUAK algorithm set: A second example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 4: Report on the design and evaluation (Release 13)*, January 2016.

[20] 3rd Generation Partnership Project (3GPP). *3GPP TS 21.111 V13.0.0 : Technical Specification; Technical Specification Group Core Network and Terminals; USIM and IC card requirements (Release 13)*, January 2016.

[21] 3rd Generation Partnership Project (3GPP). *3GPP TS 22.016 V13.0.0 : Technical Report; Technical Specification Group Services and System Aspects; International Mobile station Equipment Identities (IMEI) (Release 13)*, January 2016.

[22] 3rd Generation Partnership Project (3GPP). *3GPP TS 22.101 V14.4.0 : Technical Specification; Technical Specification Group Services and System Aspects; Service aspects; Service principles (Release 14)*, September 2016.

[23] 3rd Generation Partnership Project (3GPP). *3GPP TS 23.003 V14.1.0 : Technical Specification; Technical Specification Group Core Network and Terminals; Numbering, addressing and identification (Release 14)*, September 2016.

[24] 3rd Generation Partnership Project (3GPP). *3GPP TS 23.008 V13.6.0 : Technical Specification; Technical Specification Group Core Network and Terminals; Organisation of subscriber data (Release 13)*, September 2016.

[25] 3rd Generation Partnership Project (3GPP). *3GPP TS 23.401 V14.1.0 : Technical Specification; Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access (Release 14)*, September 2016.

[26] 3rd Generation Partnership Project (3GPP). *3GPP TS 24.008 V14.1.0 : Technical Specification; Technical Specification Group Core Network and Terminals; Mobile radio interface Layer 3 specification; Core network protocols; Stage 3 (Release 14)*, September 2016.

[27] 3rd Generation Partnership Project (3GPP). *3GPP TS 25.331 V14.1.0 : Technical Specification; Technical Specification Group Radio Access Network; Radio Resource Control (RRC); Protocol Specification (Release 14)*, December 2016.

[28] 3rd Generation Partnership Project (3GPP). *3GPP TS 31.102 V14.0.0 : Technical Specification; Technical Specification Group Core Network and Terminals; Characteristics of the Universal Subscriber Identity Module (USIM) application (Release 14)*, October 2016.

[29] 3rd Generation Partnership Project (3GPP). *3GPP TS 31.111 V14.0.0 : Technical Specification; Technical Specification Group Core Network and Terminals; Universal Subscriber Identity Module (USIM) Application Toolkit (USAT) (Release 14)*, October 2016.

[30] 3rd Generation Partnership Project (3GPP). *3GPP TS 31.130 V13.1.0 : Technical Specification; Technical Specification Group Core Network and Terminals; (U)SIM Application Programming Interface (API); (U)SIM API for Java Card (Release 13)*, October 2016.

[31] 3rd Generation Partnership Project (3GPP). *3GPP TS 33.102 V14.0.0 : Technical Specification; Technical Specification Group Services and System Aspects; 3G Security; Security architecture (Release 14)*, September 2016.

[32] 3rd Generation Partnership Project (3GPP). *3GPP TS 33.105 V13.0.0 : Technical Specification; Technical Specification Group Services and System Aspects; 3G Security; Cryptographic algorithm requirements (Release 13)*, January 2016.

[33] 3rd Generation Partnership Project (3GPP). *3GPP TS 33.220 V14.0.0 : Technical Specification; Technical Specification Group Services and System Aspects; Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA) (Release 14)*, December 2016.

[34] 3rd Generation Partnership Project (3GPP). *3GPP TS 33.401 V14.0.0 : Technical Specification; Technical Specification Group Services and System Aspects; 3GPP System Architecture Evolution (SAE); Security architecture (Release 14)*, September 2016.

[35] 3rd Generation Partnership Project (3GPP). *3GPP TS 35.206 V13.0.0 : Technical Specification; Technical Specification Group Services and System Aspects; 3G Security; Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 2: Algorithm Specification (Release 13)*, January 2016.

[36] 3rd Generation Partnership Project (3GPP). *3GPP TS 35.231 V13.0.0 : Technical Specification; Technical Specification Group Services and System Aspects; Specification of the TUAK algorithm set: A second example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 1: Algorithm specification (Release 13)*, January 2016.

[37] 3rd Generation Partnership Project (3GPP). *3GPP TS 36.300 V14.0.0 : Technical Specification; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2 (Release 14)*, September 2016.

[38] 3rd Generation Partnership Project (3GPP). *3GPP TS 36.331 V14.1.0 : Technical Specification; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification (Release 14)*, December 2016.

[39] 3rd Generation Partnership Project (3GPP). *3GPP TS 36.401 V13.2.0 : Technical Specification; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Architecture description (Release 13)*, June 2016.

[40] 3rd Generation Partnership Project (3GPP). *3GPP TS 43.020 V14.0.0 : Technical Specification; Technical Specification Group Services and System Aspects; Security related network functions (Release 14)*, September 2016.

[41] 3rd Generation Partnership Project (3GPP). *3GPP TS 55.205 V13.0.0 : Technical Specification; Technical Specification Group Services and System Aspects; Specification of the GSM-MILENAGE algorithms: An example algorithm set for the GSM authentication and key generation functions A3 and A8 (Release 13)*, January 2016.

[42] 5G Americas. *Global Mobile Subscribers and Market Share by Technology*, June 2016. [Online] Available at `http://www.5gamericas.org/en/resources/statistics/statistics-global/`.

[43] M. Abadi and B. Blanchet. Computer-assisted verification of a protocol for certified email. *Sci. Comput. Program.*, 58(1-2):3–27, 2005.

[44] M. Abadi and N. Glew. Certified email with a light on-line trusted third party: design and implementation. In D. Lassner, D. D. Roure, and A. Iyengar, editors, *Proceedings of the Eleventh International World Wide Web Conference, WWW 2002, Honolulu, Hawaii, May 7–11, 2002*, pages 387–395. ACM, 2002.

[45] The Aftenposten. *Alt om mobilspionasje-saken*, June 2015. [Online] Available at `http://mm.aftenposten.no/mobilspionasje/`.

[46] A. Agarwal, V. Shrimali, and M. L. Das. GSM security using identity-based cryptography. *CoRR*, abs/0911.0727, 2009.

[47] M. Arapinis, L. I. Mancini, E. Ritter, and M. Ryan. Privacy through pseudonymity in mobile telephony systems. In *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23–26, 2014*. The Internet Society, 2014.

[48] M. Arapinis, L. I. Mancini, E. Ritter, M. Ryan, N. Golde, K. Redon, and R. Borgaonkar. New privacy issues in mobile telephony: Fix and verification. In T. Yu, G. Danezis, and V. D. Gligor, editors, *The ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16–18, 2012*, pages 205–216. ACM, 2012.

[49] M. Arapinis, L. I. Mancini, E. Ritter, and M. D. Ryan. Analysis of privacy in mobile telephony systems. *International Journal of Information Security*, pages 1–33, 2016.

[50] G. Ateniese, A. Herzberg, H. Krawczyk, and G. Tsudik. Untraceable mobility or how to travel incognito. *Computer Networks*, 31(8):871–884, 1999.

[51] M. Barbeau and J. Robert. Perfect identity concealment in UMTS over radio access links. In *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob 2005, Montreal, Canada, August 22–24, 2005, Volume 2*, pages 72–77. IEEE, 2005.

[52] E. Barkan, E. Biham, and N. Keller. Instant ciphertext-only cryptanalysis of GSM encrypted communications. In D. Boneh, editor, *Advances in Cryptology — CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17–21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 600–616. Springer, 2003.

[53] E. Barkan, E. Biham, and N. Keller. Instant ciphertext-only cryptanalysis of GSM encrypted communication. *J. Cryptology*, 21(3):392–429, 2008.

[54] D. A. Basin, C. Cremers, and S. Meier. Provably repairing the ISO/IEC 9798 standard for entity authentication. *Journal of Computer Security*, 21(6):817–846, 2013.

[55] D. A. Basin, C. J. F. Cremers, and S. Meier. Provably repairing the ISO/IEC 9798 standard for entity authentication. In P. Degano and J. D. Guttman, editors, *Principles of Security and Trust — First International Conference, POST 2012, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2012, Tallinn, Estonia, March 24 – April 1, 2012, Proceedings*, volume 7215 of *Lecture Notes in Computer Science*, pages 129–148. Springer, 2012.

[56] M. Becher, F. C. Freiling, J. Hoffmann, T. Holz, S. Uellenbeck, and C. Wolf. Mobile security catching up? Revealing the nuts and bolts of the security of mobile devices. In *32nd IEEE Symposium on Security and Privacy, S&P 2011, 22–25 May 2011, Berkeley, California, USA*, pages 96–111. IEEE Computer Society, 2011.

[57] I. Bilogrevic, M. Jadliwala, and J.-P. Hubaux. Security issues in next generation mobile networks: LTE and femtocells. In 2nd International Femtocell Workshop, Luton, UK, June 21, 2010, 2010. Available at `https://infoscience.epfl.ch/record/149153/files/secu-LTE-femtocells-BJH-final.pdf`.

[58] B. Blanchet. An efficient cryptographic protocol verifier based on prolog rules. In *14th IEEE Computer Security Foundations Workshop (CSFW-14 2001), 11–13 June 2001, Cape Breton, Nova Scotia, Canada*, pages 82–96. IEEE Computer Society, 2001.

[59] B. Blanchet. Automatic verification of correspondences for security protocols. *Journal of Computer Security*, 17(4):363–434, 2009.

[60] B. Blanchet, B. Smyth, and V. Cheval. *ProVerif 1.88: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial*. INRIA, Paris, France, August 2013.

[61] G. Bleumer. *Encyclopedia of Cryptography and Security*, chapter Untraceability, pages 1351–1352. Springer US, 2011.

[62] R. Borgaonkar. Dirty use of USSD codes in cellular networks. In TelcoSecDay, Heidelberg, Germany, March 2013. Available at `https:`

//www.troopers.de/wp-content/uploads/2012/12/TROOPERS13-Dirty_use_of_USSD_codes_in_cellular-Ravi_Borgaonkor.pdf.

[63] R. Borgaonkar, A. Shaik, N. Asokan, V. Niemi, and J.-P. Seifert. LTE and IMSI catcher myths. In Blackhat EU, Amsterdam, Netherlands, November 2015. Available at https://www.blackhat.com/docs/eu-15/materials/eu-15-Borgaonkar-LTE-And-IMSI-Catcher-Myths.pdf.

[64] R. Borgaonkar and S. Udar. Understanding IMSI privacy. In Blackhat 2014, Las Vegas, USA, 2014. Available at https://www.isti.tu-berlin.de/fileadmin/fg214/ravi/Darshak-bh14.pdf.

[65] B. Brenninkmeijer. Catching IMSI-catcher-catchers: An effectiveness review of IMSI-catcher-catcher applications. Technical report, Radboud University, July 2016. Available at http://www.cs.ru.nl/bachelorscripties/2016/Bauke_Brenninkmeijer___4366298___Catching_IMSI-catcher-catchers.pdf.

[66] Centre for Resilient Networks and Applications (CRNA), Simula Research Laboratory. *An investigation into the claims of IMSI catchers use in Oslo in late 2014*, July 2015. [Online] Available at http://www.pst.no/media/76725/IMSI-report-SimulaResearch-Laboratory.pdf.

[67] C. Chang, J. Lee, and Y. Chang. Efficient authentication protocols of GSM. *Computer Communications*, 28(8):921–928, 2005.

[68] L. Chen, D. Gollmann, and C. J. Mitchell. Tailoring authentication protocols to match underlying mechanisms. In J. Pieprzyk and J. Seberry, editors, *Information Security and Privacy, First Australasian Conference, ACISP'96, Wollongong, NSW, Australia, June 24–26, 1996, Proceedings*, volume 1172 of *Lecture Notes in Computer Science*, pages 121–133. Springer, 1996.

[69] L. Chen and M. Ryan. Attack, solution and verification for shared authorisation data in TCG TPM. In P. Degano and J. D. Guttman, editors, *Formal Aspects in Security and Trust, 6th International Workshop, FAST 2009, Eindhoven, The Netherlands, November 5–6, 2009, Revised Selected Papers*, volume 5983 of *Lecture Notes in Computer Science*, pages 201–216. Springer, 2009.

[70] Y. J. Choi and S. Kim. An improvement on privacy and authentication in GSM. In C. H. Lim and M. Yung, editors, *Information Security Applications, 5th International Workshop, WISA 2004, Jeju Island, Korea, August 23–25, 2004, Revised Selected Papers*, volume 3325 of *Lecture Notes in Computer Science*, pages 14–26. Springer, 2004.

[71] H. Choudhury, B. Roychoudhury, and D. K. Saikia. Enhancing user identity privacy in LTE. In G. Min, Y. Wu, L. C. Liu, X. Jin, S. A. Jarvis, and A. Y. Al-Dubai, editors, *11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2012, Liverpool, United Kingdom, June 25–27, 2012*, pages 949–957. IEEE Computer Society, 2012.

[72] A. Cooper, H. Tschofenig, B. Aboba, J. Peterson, J. Morris, M. Hansen, and R. Smith. Privacy considerations for Internet protocols. RFC 6973, RFC Editor, July 2013. Available at `http://www.rfc-editor.org/info/rfc6973`.

[73] A. Dabrowski, N. Pianta, T. Klepp, M. Mulazzani, and E. R. Weippl. IMSI-catch me if you can: IMSI-catcher-catchers. In C. N. P. Jr., A. Hahn, K. R. B. Butler, and M. Sherr, editors, *Proceedings of the 30th Annual Computer Security Applications Conference, ACSAC 2014, New Orleans, LA, USA, December 8–12, 2014*, pages 246–255. ACM, 2014.

[74] J. P. Degabriele, A. Lehmann, K. G. Paterson, N. P. Smart, and M. Strefler. On the joint security of encryption and signature in EMV. In O. Dunkelman, editor, *Topics in Cryptology — CT-RSA 2012 — The Cryptographers' Track at the RSA Conference 2012, San Francisco, CA, USA, February 27 – March 2, 2012. Proceedings*, volume 7178 of *Lecture Notes in Computer Science*, pages 116–135. Springer, 2012.

[75] S. Delaune, S. Kremer, and M. Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, 17(4):435–487, 2009.

[76] Y. Deng, H. Fu, X. Xie, J. Zhou, Y. Zhang, and J. Shi. A novel 3GPP SAE authentication and key agreement protocol. In *2009 IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC 2009), Beijing, China, 6–8 November, 2009*, pages 557–561. IEEE, 2009.

[77] C. K. Dimitriadis. Improving mobile core network security with honeynets. *IEEE Security & Privacy*, 5(4):40–47, 2007.

[78] D. Dolev and A. C. Yao. On the security of public key protocols. *IEEE Trans. Information Theory*, 29(2):198–207, 1983.

[79] M. Dupré. Process to control a Subscriber Identity Module (SIM) in mobile phone system. US Patent Office, Available at `https://www.google.com/patents/US6690930`, February 2004. US Patent 6690930 B1, Filing date — 25 May, 1999.

[80] T. Engel. Locating mobile phones using Signalling System 7. In 25th Chaos Communication Congress (25C3), December 2008. Availalable at https://events.ccc.de/congress/2008/Fahrplan/attachments/1262_25c3-locating-mobile-phones.pdf.

[81] Ericsson. Enhancements to GSM/UMTS AKA. 3GPP TSG SA WG3 Security, Oct. 6–10 2003. Document reference — S3-030542, Available at http://www.3gpp.org/ftp/tsg_sa/wg3_security/tsgs3_30_povoa/docs/pdf/S3-030542.pdf.

[82] Ericsson. On the introduction and use of UMTS AKA in GSM. 3GPP TSG SA WG3 Security, July 6–9 2004. Document reference — S3-040534. Available at ftp://www.3gpp.org/tsg_sa/WG3_Security/TSGS3_34_Acapulco/Docs/PDF/S3-040534.pdf.

[83] European Telecommunications Standards Institute (ETSI). *ETSI TS 100 530 V7.0.0 : Technical Specification; Digital cellular telecommunications system (Phase 2+); Location registration procedures (GSM 03.12 version 7.0.0 Release 1998)*, August 1999.

[84] European Telecommunications Standards Institute (ETSI). *GSM 02.17 V8.0.0 : Technical Specification; Digital cellular telecommunications system (Phase 2+); Subscriber Identity Modules (SIM); Functional characteristics (GSM 02.17 version 8.0.0 Release 1999)*, November 1999.

[85] European Telecommunications Standards Institute (ETSI). *ETSI TS 100 508 V7.2.0 : Technical Specification; Digital cellular telecommunications system (Phase 2+); International Mobile station Equipment Identities (IMEI) (GSM 02.16 version 7.2.0 Release 1998)*, June 2000.

[86] European Telecommunications Standards Institute (ETSI). *GSM 01.04 V8.0.0 : Technical Report; Digital cellular telecommunications system (Phase 2+); Abbreviations and acronyms (GSM 01.04 version 8.0.0 release 1999)*, May 2000.

[87] European Telecommunications Standards Institute (ETSI). *TS 100 522 V7.1.0 : Technical Specification; Digital cellular telecommunications system (Phase 2+); Network architecture (GSM 03.02 version 7.1.0 Release 1998)*, January 2000.

[88] European Telecommunications Standards Institute (ETSI). *ETSI TS 102 221 V8.2.0 : Technical Specification; Smart Cards; UICC–Terminal interface; Physical and logical characteristics (Release 8)*, June 2009.

[89] European Telecommunications Standards Institute (ETSI). *ETSI TS 102 223 V12.1.0 : Technical Specification; Smart Cards; Card Application Toolkit (CAT) (Release 12)*, September 2014.

[90] A. Fanian, M. Berenjkoub, and T. A. Gulliver. A new mutual authentication protocol for GSM networks. In *Proceedings of the 22nd Canadian Conference on Electrical and Computer Engineering, CCECE 2009, Delta St. John's Hotel and Conference Centre, St. John's, Newfoundland, Canada, May 3–6, 2009*, pages 798–803. IEEE, 2009.

[91] A. Fanian, M. Berenjkoub, and T. A. Gulliver. A TESLA-based mutual authentication protocol for GSM networks. *The ISC International Journal of Information Security*, 1(1), 2009.

[92] F. Farhat, S. Salimi, and A. Salahi. Private identification, authentication and key agreement protocol with security mode setup. *IACR Cryptology ePrint Archive*, 2011:45, 2011.

[93] D. Forsberg, G. Horn, W.-D. Moeller, and V. Niemi. *LTE Security*. John Wiley & Sons, 2010.

[94] P. Fouque, C. Onete, and B. Richard. Achieving better privacy for the 3GPP AKA protocol. *IACR Cryptology ePrint Archive*, 2016:480, 2016.

[95] S. Gibbs. SS7 hack explained: What can you do about it? The Guardian, Apr. 19 2016. Available at `https://www.theguardian.com/technology/2016/apr/19/ss7-hack-explained-mobile-phone-vulnerability-snooping-texts-calls`.

[96] I. Goldberg, D. Wagner, and L. Green. The (real-time) cryptanalysis of A5/2. Presented at the Rump session of Crypto'99, August 1999. Available at `https://people.eecs.berkeley.edu/~daw/tmp/a52-slides.ps`.

[97] N. Golde, K. Redon, and R. Borgaonkar. Weaponizing femtocells: The effect of rogue devices on mobile telecommunications. In *19th Annual Network and Distributed System Security Symposium, NDSS 2012, San Diego, California, USA, February 5–8, 2012*. The Internet Society, 2012.

[98] N. Golde, K. Redon, and J. Seifert. Let me answer that for you: Exploiting broadcast information in cellular networks. In S. T. King, editor, *Proceedings of the 22th USENIX Security Symposium, Washington, DC, USA, August 14–16, 2013*, pages 33–48. USENIX Association, 2013.

[99] O. Goldreich. *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, 2004.

[100] D. Goodin. Low-cost IMSI catcher for 4G/LTE networks tracks phones precise locations, October 2015. [Online] Available at `https://arstechnica.com/civis/viewtopic.php?f=2&t=1298409`.

[101] C. Han and H. Choi. Security analysis of handover key management in 4G LTE/SAE networks. *IEEE Trans. Mob. Comput.*, 13(2):457–468, 2014.

[102] A. Herzberg, H. Krawczyk, and G. Tsudik. On travelling incognito. In *First Workshop on Mobile Computing Systems and Applications, WMCSA 1994, Santa Cruz, CA, USA, December 8–9, 1994*, pages 205–211. IEEE Computer Society, 1994.

[103] International Organisation for Standardisation, Genève, Switzerland. *ISO/IEC 9798-4: 1999; Information technology — Security techniques — Entity authentication — Part 4: Mechanisms using a cryptographic check function*, 2nd edition, 1999.

[104] International Organisation for Standardisation, Genève, Switzerland. *ISO/IEC 19772:2009; Information technology — Security techniques — Authenticated encryption mechanisms*, February 2009.

[105] International Organisation for Standardisation, Genève, Switzerland. *ISO/IEC 9798-4:1999/Cor 1:2009; Technical Corrigendum 1*, 2009.

[106] International Organisation for Standardisation (ISO) and International Electrotechnical Commission (IEC). *ISO/IEC 7816-3:2006; International Standard; Identification cards—Integrated circuit cards—Part 3: Cards with contacts—Electrical interface and transmission protocols*, November 2006.

[107] International Organisation for Standardisation (ISO) and International Electrotechnical Commission (IEC). *ISO/IEC 9797-1:2011; Information technology — Security techniques — Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher*, 2011.

[108] W. Juang and J. Wu. Efficient 3GPP authentication and key agreement with robust user privacy protection. In *IEEE Wireless Communications and Networking Conference, WCNC 2007, Hong Kong, China, 11–15 March, 2007*, pages 2720–2725. IEEE, 2007.

[109] M. S. A. Khan and C. J. Mitchell. Another look at privacy threats in 3G mobile telephony. In W. Susilo and Y. Mu, editors, *Information Security and Privacy — 19th Australasian Conference, ACISP 2014, Wollongong, NSW, Australia, July 7–9, 2014. Proceedings*, volume 8544 of *Lecture Notes in Computer Science*, pages 386–396. Springer, 2014.

[110] M. S. A. Khan and C. J. Mitchell. Improving air interface user privacy in mobile telephony. In L. Chen and S. Matsuo, editors, *Security Standardisation Research — Second International Conference, SSR 2015, Tokyo, Japan, December 15–16, 2015, Proceedings*, volume 9497 of *Lecture Notes in Computer Science*, pages 165–184. Springer, 2015.

[111] M. S. A. Khan and C. J. Mitchell. Retrofitting mutual authentication to GSM using RAND hijacking. In G. Barthe, E. P. Markatos, and P. Samarati, editors, *Security and Trust Management — 12th International Workshop, STM 2016, Heraklion, Crete, Greece, September 26–27, 2016, Proceedings*, volume 9871 of *Lecture Notes in Computer Science*, pages 17–31. Springer, 2016.

[112] P. C. Kocher. On certificate revocation and validation. In R. Hirschfeld, editor, *Financial Cryptography, Second International Conference, FC'98, Anguilla, British West Indies, February 23–25, 1998, Proceedings*, volume 1465 of *Lecture Notes in Computer Science*, pages 172–177. Springer, 1998.

[113] G. M. Køien. Privacy enhanced mutual authentication in LTE. In *9th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob 2013, Lyon, France, October 7–9, 2013*, pages 614–621. IEEE Computer Society, 2013.

[114] G. M. Køien and V. A. Oleshchuk. *Aspects of Personal Privacy in Communications: Problems, Technology and Solutions*. Denmark: River Publishers, 2013.

[115] S. Kremer, M. Ryan, and B. Smyth. Election verifiability in electronic voting protocols. In D. Gritzalis, B. Preneel, and M. Theoharidou, editors, *Computer Security — ESORICS 2010, 15th European Symposium on Research in Computer Security, Athens, Greece, September 20–22, 2010. Proceedings*, volume 6345 of *Lecture Notes in Computer Science*, pages 389–404. Springer, 2010.

[116] K. P. Kumar, G. Shailaja, A. Kavitha, and A. Saxena. Mutual authentication and key agreement for GSM. In *International Conference on Mobile Business (ICMB 2006), Copenhagen, Denmark, June 26–27, 2006*, pages 25–28. IEEE Computer Society, 2006.

[117] D. F. Kune, J. Kölndorfer, N. Hopper, and Y. Kim. Location leaks over the GSM air interface. In *19th Annual Network and Distributed System Security Symposium, NDSS 2012, San Diego, California, USA, February 5–8, 2012*. The Internet Society, 2012.

[118] C. Lee, M. Hwang, and W. Yang. Enhanced privacy and authentication for the global system for mobile communications. *Wireless Networks*, 5(4):231–243, 1999.

[119] C. C. Lee, M. S. Hwang, and W. P. Yang. Extension of authentication protocol for GSM. *IEE Proceedings – Communications*, 150(2):91–95, April 2003.

[120] M. Lee, N. P. Smart, B. Warinschi, and G. J. Watson. Anonymity guarantees of the UMTS/LTE authentication and connection protocol. *Int. J. Inf. Sec.*, 13(6):513–527, 2014.

[121] C. Liu. Worldwide Internet and mobile users: eMarketer's updated estimates for 2015, August 2015. [Online] Available at `https://insights.ap.org/uploads/images/eMarketer_Estimates_2015.pdf`.

[122] C. Lo and Y. Chen. Secure communication mechanisms for GSM networks. *IEEE Transactions on Consumer Electronics*, 45(4):1074–1080, November 1999.

[123] I. Marsden and P. Marshall. Multi IMSI system and method. US Patent Office, Avaiable at `http://www.google.com/patents/US20140051423`, February 2014. Patent Application US 13/966,350, Filing date — 14 August, 2013.

[124] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[125] U. Meyer and S. Wetzel. A man-in-the-middle attack on UMTS. In M. Jakobsson and A. Perrig, editors, *Proceedings of the 2004 ACM Workshop on Wireless Security, Philadelphia, PA, USA, October 1, 2004*, pages 90–97. ACM, 2004.

[126] U. Meyer and S. Wetzel. On the impact of GSM encryption and man-in-the-middle attacks on the security of interoperating GSM/UMTS networks. In *Proceedings of the IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2004, Barcelona, Spain, September 5–8, 2004*, pages 2876–2883. IEEE, 2004.

[127] C. Mitchell, J. Brown, L. Chen, D. Goj, D. Gollmann, Y.-F. Han, N. Jefferies, M. Walker, and D. Youngs. *LINK 3GS3 Technical Report 2: Security Mechanisms for Third Generation Systems*. Vodafone Ltd., GPT Ltd.,

and ISG, Royal Holloway, University of London, May 15 1996. Available at http://www.chrismitchell.net/3GS3/TR2_pdf.ZIP.

[128] C. J. Mitchell. Security in future mobile networks. In *Proceedings of the Second International Workshop on Mobile Multi-Media Communications (MoMuC-2), Bristol, UK, April, 1995*, 1995.

[129] C. J. Mitchell. Making serial number based authentication robust against loss of state. *ACM Operating Systems Review*, 34(3):56–59, 2000.

[130] C. J. Mitchell. The security of the GSM air interface protocol. Technical Report RHUL-MA-2001-3, Mathematics Department, Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK, August 2001. Available at http://www.ma.rhul.ac.uk/techreports.

[131] M. Myers. Revocatoin: Options and challenges. In R. Hirschfeld, editor, *Financial Cryptography, Second International Conference, FC'98, Anguilla, British West Indies, February 23–25, 1998, Proceedings*, volume 1465 of *Lecture Notes in Computer Science*, pages 165–171. Springer, 1998.

[132] NGMN Alliance. *NGMN 5G White Paper*, February 2015. [Online] Available at https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf.

[133] K. Nohl. Mobile self-defense. In 31st Chaos Communication Congress (31C3), 2014. Available at https://events.ccc.de/congress/2014/Fahrplan/system/attachments/2493/original/Mobile_Self_Defense-Karsten_Nohl-31C3-v1.pdf.

[134] Ofcom. *The Communications Market Report*, August 2016. [Online] Available at https://www.ofcom.org.uk/__data/assets/pdf_file/0024/26826/cmr_uk_2016.pdf.

[135] J. Ooi. IMSI catchers and mobile security. Technical report, University of Pennsylvania, April 2015. Available at https://www.cis.upenn.edu/current-students/undergraduate/courses/documents/EAS499Honors-IMSICatchersandMobileSecurity-V18F-1.pdf.

[136] P. S. Pagliusi. A contemporary foreword on GSM security. In G. I. Davida, Y. Frankel, and O. Rees, editors, *Infrastructure Security, International Conference, InfraSec 2002, Bristol, UK, October 1–3, 2002, Proceedings*, volume 2437 of *Lecture Notes in Computer Science*, pages 129–144. Springer, 2002.

[137] A. Perrig, R. Canetti, J. D. Tygar, and D. Song. The TESLA broadcast authentication protocol. *RSA Cryptobytes*, 5(2), 2002.

[138] F. Piper and M. Walker. Cryptographic solutions for voice telephony and GSM. *Network Security*, 1998(12):14–19, December 1998.

[139] S. P. Rao, S. Holtmanns, I. Oliver, and T. Aura. Unblocking stolen mobile devices using SS7-MAP vulnerabilities: Exploiting the relationship between IMEI and IMSI for EIR access. In *2015 IEEE TrustCom/BigDataSE/ISPA, Helsinki, Finland, August 20-22, 2015, Volume 1*, pages 1171–1176. IEEE, 2015.

[140] S. P. Rao, I. Oliver, S. Holtmanns, and T. Aura. We know where you are! In N. Pissanidis, H. Roigas, and M. Veenendaal, editors, *8th International Conference on Cyber Conflict, CyCon 2016, Tallinn, Estonia, May 31 – June 3, 2016*, pages 277–293. IEEE, 2016.

[141] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[142] D. Samfat and R. Molva. A method providing identity privacy to mobile users during authentication. In *First Workshop on Mobile Computing Systems and Applications, WMCSA 1994, Santa Cruz, CA, USA, December 8–9, 1994*, pages 196–199. IEEE Computer Society, 1994.

[143] D. Samfat, R. Molva, and N. Asokan. Untraceability in mobile networks. In B. Awerbuch and D. Duchamp, editors, *MOBICOM '95, Proceedings of the First Annual International Conference on Mobile Computing and Networking, Berkeley, CA, USA, November 13–15, 1995*, pages 26–36. ACM, 1995.

[144] B. Sattarzadeh, M. Asadpour, and R. Jalili. Improved user identity confidentiality for UMTS mobile networks. In *Fourth European Conference on Universal Multiservice Networks (ECUMN 2007), 14–16 February 2007, Toulouse, France*, pages 401–409. IEEE Computer Society, 2007.

[145] J. Scahill and J. Begley. The great SIM heist—how spies stole the keys to the encryption castle, February 2015. [Online] Available at `https://theintercept.com/2015/02/19/great-simheist/`.

[146] P. Schneider and G. Horn. Towards 5G security. In *2015 IEEE TrustCom/BigDataSE/ISPA, Helsinki, Finland, August 20–22, 2015, Volume 1*, pages 1165–1170. IEEE, 2015.

[147] A. Shaik, J. Seifert, R. Borgaonkar, N. Asokan, and V. Niemi. Practical attacks against privacy and availability in 4G/LTE mobile communication systems. In *23nd Annual Network and Distributed System Security Symposium, NDSS 2016, San Diego, California, USA, February 21–24, 2016*. The Internet Society, 2016.

[148] A. Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and D. Chaum, editors, *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19–22, 1984, Proceedings*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984.

[149] B. Smyth, M. Ryan, S. Kremer, and M. Kourjieh. Towards automatic analysis of election verifiability properties. In A. Armando and G. Lowe, editors, *Automated Reasoning for Security Protocol Analysis and Issues in the Theory of Security — Joint Workshop, ARSPA-WITS 2010, Paphos, Cyprus, March 27–28, 2010. Revised Selected Papers*, volume 6186 of *Lecture Notes in Computer Science*, pages 146–163. Springer, 2010.

[150] A. Soltani and C. Timberg. Tech firm tries to pull back curtain on surveillance efforts in Washington, September 2014. [Online] Available at `http://wapo.st/1qgzImt`.

[151] D. Strobel. IMSI catcher. Technical report, Ruhr-Universität Bochum, July 2007. Available at `https://www.emsec.rub.de/media/crypto/attachments/files/2011/04/imsi_catcher.pdf`.

[152] K. Sung, B. N. Levine, and M. Liberatore. Location privacy without carrier cooperation. In *IEEE workshop on Mobile Security Technologies, MoST, San Jose, CA, USA, May 17, 2014*. IEEE, 2014.

[153] J. Tagg and A. J. Campbell. Identity management for mobile devices. US Patent Office, Available at `http://www.google.com/patents/US20120309374`, December 2012. Patent Application US 13/151,942, Filing date — 02 June, 2011.

[154] C. Tang. *Modeling and Analysis of Mobile Telephony Protocols*. PhD thesis, Stevens Institute of Technology, 2013. Available at `http://www.cs.stevens.edu/~Naumann/publications/ChunyuTangDiss.pdf`.

[155] C. Tang, D. A. Naumann, and S. Wetzel. Symbolic analysis for security of roaming protocols in mobile networks [extended abstract]. In M. Rajarajan, F. Piper, H. Wang, and G. Kesidis, editors, *Security and Privacy in Communication Networks — 7th International ICST Conference, SecureComm 2011, London, UK,*

September 7–9, 2011, Revised Selected Papers, volume 96 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 480–490. Springer, 2011.

[156] C. Tang, D. A. Naumann, and S. Wetzel. Analysis of authentication and key establishment in inter-generational mobile telephony. In *10th IEEE International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing, HPCC/EUC 2013, Zhangjiajie, China, November 13–15, 2013*, pages 1605–1614. IEEE, 2013.

[157] Telecommunication Standardisation Sector of ITU. *ITU-T E.212—The international identification plan for public networks and subscriptions*, 2008.

[158] Telecommunication Standardisation Sector of ITU. *ITU-T E.164—The international public telecommunication numbering plan*, November 2010.

[159] C. Timberg. Feds to study illegal use of spy gear, August 2014. [Online] Available at `https://www.washingtonpost.com/news/the-switch/wp/2014/08/11/feds-to-study-illegal-use-of-spy-gear/`.

[160] M. Toorani and A. Beheshti. Solutions to the GSM security weaknesses. In *Next Generation Mobile Applications, Services and Technologies, 2008, NGMAST'08, The Second International Conference on*, pages 576–581. IEEE, 2008.

[161] J. Tsay and S. F. Mjølsnes. A vulnerability in the UMTS and LTE authentication and key agreement protocols. In I. V. Kotenko and V. A. Skormin, editors, *Computer Network Security — 6th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security, MMM-ACNS 2012, St. Petersburg, Russia, October 17-19, 2012. Proceedings*, volume 7531 of *Lecture Notes in Computer Science*, pages 65–76. Springer, 2012.

[162] N. Valtteri and K. Nyberg. *UMTS Security.* John Wiley & Sons Limited, 2003.

[163] F. van den Broek, R. Verdult, and J. de Ruiter. Defeating IMSI catchers. In I. Ray, N. Li, and C. Kruegel, editors, *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12–16, 2015*, pages 340–351. ACM, 2015.

[164] Various Contributors. AIMSICD — Android-IMSI-Catcher-Detector. [online] Available at `https://github.com/CellularPrivacy/Android-IMSI-Catcher-Detector`.

[165] Various Contributors. Darshak Framework. [online] Available at `https://github.com/darshakframework/darshak`.

[166] Various Contributors. OpenBTS — An open source cellular infrastructure. [online] Available at `http://openbts.org`.

[167] Various Contributors. OpenLTE — An open source 3GPP LTE implementation. [online] Available at `https://sourceforge.net/projects/openlte/`.

[168] Various Contributors. OsmoBSC — An open source GSM base statiom controller. [online] Available at `http://osmocom.org/projects/osmobsc`.

[169] Various Contributors. Osmocom SIMtrace. [Online] Available at `http://bb.osmocom.org/trac/wiki/SIMtrace`.

[170] Various Contributors. Snoopsnitch — SRLabs open source project. [online] Available at `https://opensource.srlabs.de/projects/snoopsnitch`.

[171] Various Contributors. srsLTE — An open source 3GPP LTE library. [online] Available at `https://github.com/srsLTE/srsLTE`.

[172] S. Vaudenay. Security flaws induced by CBC padding — applications to SSL, IPSEC, WTLS ... In L. R. Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 – May 2, 2002, Proceedings*, volume 2332 of *Lecture Notes in Computer Science*, pages 534–546. Springer-Verlag, Berlin, 2002.

[173] Vodafone. Cipher key separation for A/Gb security enhancements. 3GPP TSG SA WG3 Security, July 15–18 2003. Document reference — S3-030463. Available at `ftp://www.3gpp.org/tsg_sa/WG3_Security/TSGS3_29_SanFran/Docs/PDF/S3-030463.pdf`.

[174] T. Y. C. Woo and S. S. Lam. A semantic model for authentication protocols. In *1993 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, USA, May 24–26, 1993*, pages 178–194. IEEE Computer Society, 1993.

[175] L. Xiehua and W. Yongjun. Security enhanced authentication and key agreement protocol for LTE/SAE network. In *The 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), Wuhan, China, 23–25 September, 2011*, pages 1–4. IEEE, 2011.

[176] M. Zhang and Y. Fang. Security analysis and enhancements of 3GPP authentication and key agreement protocol. *IEEE Trans. Wireless Communications*, 4(2):734–742, 2005.

[177] J. Zhu and J. Ma. A new authentication scheme with anonymity for wireless environments. *IEEE Trans. Consumer Electronics*, 50(1):231–235, 2004.

Appendix A

GSM Analysis

A.1 ProVerif Model of Modified GSM AKA

```
1  (* Communication Channels *)
2  free pubChannel: channel.
3  free secureChannel: channel [private].

5  (* Type declaration *)
6          type key.
7          type sessionKey.
8          type anonymityKey.
9          type mac.
10         type ident.
11         type resp.
12         type msgHdr.

14         const ID: msgHdr.
15         const AV_REQ: msgHdr.
16         const AV: msgHdr.
17         const CHALLENGE: msgHdr.
18         const SRES: msgHdr.

20 (* Private data which secrecy we are interested in *)
21         free sqn: bitstring [private].

23 (* Cryptographic functions *)
24         fun a3(bitstring, mac, key): resp.
25         fun a8(bitstring, mac, key): sessionKey.
26         fun f1(bitstring, key): mac.
```

```
27          fun f5(mac, key): anonymityKey.
28          fun encrypt(bitstring, anonymityKey): bitstring.

30  (* Reduction rule *)
31          reduc forall m: bitstring, k: anonymityKey; decrypt(encrypt(
                m, k), k) = m.

33  (* Store identity and key pair *)
34          table keys(ident, key).

36          event begSN(ident, sessionKey).
37          event endSN(ident, sessionKey).
38          event begMS(ident, sessionKey).
39          event endMS(ident, sessionKey).

41  (* Reachability / Secrecy Query *)
42          query attacker(sqn).

44  (* Authentication Query *)
45  (* To verify that SN authenticates MS *)
46          query x1: ident, x2: sessionKey; event(endSN(x1, x2)) ==>
                event(begSN(x1, x2)).
47  (* To verify that MS authenticates SN *)
48          query x1: ident, x2: sessionKey; event(endMS(x1, x2)) ==>
                event(begMS(x1, x2)).

50  (* Mobile Station Sub-Process *)
51  let processMS=
52          new imsi_ms: ident;
53          new ki: key;
54          insert keys(imsi_ms, ki);
55          out(pubChannel, (ID, imsi_ms));
56          in(pubChannel, (=CHALLENGE, enc_sqn_ms: bitstring, mac_ms:
                mac));
57          let ak_ms: anonymityKey = f5(mac_ms, ki) in
58          let sqn_ms: bitstring = decrypt(enc_sqn_ms, ak_ms) in
59          if f1(sqn_ms, ki) = mac_ms then (
60                  let res_ms: resp = a3(enc_sqn_ms, mac_ms, ki) in
61                  let kc_ms: sessionKey = a8(enc_sqn_ms, mac_ms, ki)
                        in
62                  event endMS(imsi_ms, kc_ms);
63                  event begSN(imsi_ms, kc_ms);
```

```
64                  out(pubChannel, (SRES, res_ms))) else (
65          new d_res: resp;
66          out(pubChannel, (SRES, d_res))).

68  (* Serving Network Sub-Process *)
69  let processSN=
70          in(pubChannel, (=ID, imsi_sn: ident));
71          out(secureChannel, (AV_REQ, imsi_sn));
72          in(secureChannel, (=AV, imsi_hn_sn: ident, enc_sqn_sn:
                bitstring,
73          mac_sn: mac, xres_sn: resp, kc_sn: sessionKey));
74          event begMS(imsi_hn_sn, kc_sn);
75          out(pubChannel, (CHALLENGE, enc_sqn_sn, mac_sn));
76          in(pubChannel, (=SRES, res_sn: resp));
77          if res_sn = xres_sn then
78          event endSN(imsi_hn_sn, kc_sn).

80  (* Home Network Sub-Process *)
81  let processHN=
82          in(secureChannel, (=AV_REQ, imsi_hn: ident));
83          get keys(=imsi_hn, ki_hn) in
84          let mac_hn: mac = f1(sqn, ki_hn) in
85          let ak_hn: anonymityKey = f5(mac_hn, ki_hn) in
86          let enc_sqn_hn: bitstring = encrypt(sqn, ak_hn) in
87          let xres_hn: resp = a3(enc_sqn_hn, mac_hn, ki_hn) in
88          let kc_hn: sessionKey = a8(enc_sqn_hn, mac_hn, ki_hn) in
89          out(secureChannel, (AV, imsi_hn, enc_sqn_hn, mac_hn, xres_hn
                , kc_hn)).

91  (* Main Process *)
92  process
93          ((!processMS) | processSN | processHN)
```

A.2 ProVerif Model of GSM AKA

```
1  (* Communication Channels *)
2  free pubChannel: channel.
3  free secureChannel: channel [private].

5  (* Type declaration *)
6          type key.
7          type sessionKey.
8          type nonce.
9          type ident.
10         type resp.
11         type msgHdr.

13         const ID: msgHdr.
14         const AV_REQ: msgHdr.
15         const AV: msgHdr.
16         const CHALLENGE: msgHdr.
17         const RES: msgHdr.

19 (* Cryptographic functions *)
20         fun a3(nonce, key): resp.
21         fun a8(nonce, key): sessionKey.

23 (* Store identity and key pairs *)
24         table keys(ident, key).

26         event begSN(ident, sessionKey).
27         event endSN(ident, sessionKey).
28         event begMS(ident, sessionKey).
29         event endMS(ident, sessionKey).

31 (* Authentication Query *)
32 (* To verify that SN authenticates MS *)
33         query x1: ident, x2: sessionKey; event(endSN(x1, x2)) ==>
               event(begSN(x1, x2)).
34 (* To verify that MS authenticates SN *)
35         query x1: ident, x2: sessionKey; event(endMS(x1, x2)) ==>
               event(begMS(x1, x2)).

37 (* Mobile Station Sub-Process *)
38 let processMS=
```

```
39          new imsi_ms: ident;
40          new ki: key;
41          insert keys(imsi_ms, ki);
42          out(pubChannel, (ID, imsi_ms));
43          in(pubChannel, (=CHALLENGE, rand_ms: nonce));
44          let res_ms: resp = a3(rand_ms, ki) in
45          let kc_ms: sessionKey = a8(rand_ms, ki) in
46          event endMS(imsi_ms, kc_ms);
47          event begSN(imsi_ms, kc_ms);
48          out(pubChannel, (RES, res_ms)).

50  (* Serving Network Sub-Process *)
51  let processSN=
52          in(pubChannel, (=ID, imsi_sn: ident));
53          out(secureChannel, (AV_REQ, imsi_sn));
54          in(secureChannel, (=AV, imsi_hn_sn: ident, rand_sn: nonce,
                xres_sn: resp, kc_sn: sessionKey));
55          event begMS(imsi_hn_sn, kc_sn);
56          out(pubChannel, (CHALLENGE, rand_sn));
57          in(pubChannel, (=RES, res_sn: resp));
58          if res_sn = xres_sn then
59          event endSN(imsi_hn_sn, kc_sn).

61  (* Home Network Sub-Process *)
62  let processHN=
63          in(secureChannel, (=AV_REQ, imsi_hn: ident));
64          new rand_hn: nonce;
65          get keys(=imsi_hn, ki_hn) in
66          let xres_hn: resp = a3(rand_hn, ki_hn) in
67          let kc_hn: sessionKey = a8(rand_hn, ki_hn) in
68          out(secureChannel, (AV, imsi_hn, rand_hn, xres_hn, kc_hn)).

70  (* Main Process *)
71  process
72          ((!processMS) | processSN | processHN)
```

A.3 ProVerif Output of Model Execution

A.3.1 Modified GSM AKA

```
1  (* Verification of Authentication Property: SN Authentication *)

3  -- Query event(endMS(x1,x2)) ==> event(begMS(x1,x2))
4  Completing...
5  Starting query event(endMS(x1,x2)) ==> event(begMS(x1,x2))
6  goal reachable: begin(begMS(imsi_ms[!1 = endsid_1016],a8(encrypt(sqn
       [],f5(f1(sqn[],ki[!1 = endsid_1016]),ki[!1 = endsid_1016])),f1(
       sqn[],ki[!1 = endsid_1016]),ki[!1 = endsid_1016])), kc_sn = a8(
       encrypt(sqn[],f5(f1(sqn[],ki[!1 = endsid_1016]),ki[!1 =
       endsid_1016])),f1(sqn[],ki[!1 = endsid_1016]),ki[!1 = endsid_1016
       ]), xres_sn = a3(encrypt(sqn[],f5(f1(sqn[],ki[!1 = endsid_1016]),
       ki[!1 = endsid_1016])),f1(sqn[],ki[!1 = endsid_1016]),ki[!1 =
       endsid_1016]), mac_sn = f1(sqn[],ki[!1 = endsid_1016]),
       enc_sqn_sn = encrypt(sqn[],f5(f1(sqn[],ki[!1 = endsid_1016]),ki
       [!1 = endsid_1016])), imsi_hn_sn = imsi_ms[!1 = endsid_1016],
       imsi_sn = imsi_sn_1017, @sid_417 = @sid_1018, @occ21 = @occ_cst)
       && attacker(imsi_sn_1017) -> end(endsid_1016,endMS(imsi_ms[!1 =
       endsid_1016],a8(encrypt(sqn[],f5(f1(sqn[],ki[!1 = endsid_1016]),
       ki[!1 = endsid_1016])),f1(sqn[],ki[!1 = endsid_1016]),ki[!1 =
       endsid_1016]))))

8  RESULT event(endMS(x1,x2)) ==> event(begMS(x1,x2)) is true.

10 (* Verification of Authentication Property: MS Authentication *)

12 -- Query event(endSN(x1_1029,x2_1030)) ==> event(begSN(x1_1029,
       x2_1030))
13 Completing...
14 Starting query event(endSN(x1_1029,x2_1030)) ==> event(begSN(x1_1029
       ,x2_1030))
15 goal reachable: begin(begSN(imsi_ms[!1 = @sid_1782],a8(encrypt(sqn
       [],f5(f1(sqn[],ki[!1 = @sid_1782]),ki[!1 = @sid_1782])),f1(sqn[],
       ki[!1 = @sid_1782]),ki[!1 = @sid_1782]))) -> end(endSN(imsi_ms[!1
        = @sid_1782],a8(encrypt(sqn[],f5(f1(sqn[],ki[!1 = @sid_1782]),ki
       [!1 = @sid_1782])),f1(sqn[],ki[!1 = @sid_1782]),ki[!1 = @sid_1782
       ])))
```

```
17 RESULT event(endSN(x1_1029, x2_1030)) ==> event(begSN(x1_1029, x2_1030
       )) is true.

19 (* Verification of Secrecy Property *)

21 -- Query not attacker(sqn[])
22 Completing...
23 Starting query not attacker(sqn[])

25 RESULT not attacker(sqn[]) is true.
```

A.3.2 Original GSM AKA

```
1  (* Verification of Authentication Property: MS Authentication *)

3  -- Query event(endSN(x1_575,x2_576)) ==> event(begSN(x1_575,x2_576))
4  Completing...
5  Starting query event(endSN(x1_575,x2_576)) ==> event(begSN(x1_575,
       x2_576))
6  goal reachable: begin(begSN(imsi_ms[!1 = @sid_1040],a8(rand_hn[
       imsi_hn = imsi_ms[!1 = @sid_1040],!1 = @sid_1041],ki[!1 =
       @sid_1040]))) -> end(endSN(imsi_ms[!1 = @sid_1040],a8(rand_hn[
       imsi_hn = imsi_ms[!1 = @sid_1040],!1 = @sid_1041],ki[!1 =
       @sid_1040])))

8  RESULT event(endSN(x1_575,x2_576)) ==> event(begSN(x1_575,x2_576))
       is true.

10 (* Verification of Authentication Property: SN Authentication *)

12 -- Query event(endMS(x1,x2)) ==> event(begMS(x1,x2))
13 Completing...
14 Starting query event(endMS(x1,x2)) ==> event(begMS(x1,x2))
15 goal reachable: attacker(rand_ms_548) -> end(endsid_549,endMS(
       imsi_ms[!1 = endsid_549],a8(rand_ms_548,ki[!1 = endsid_549])))
16 Abbreviations:
17 imsi = imsi_ms[!1 = endsid_557]
18 ki_560 = ki[!1 = endsid_557]

20 1. We assume as hypothesis that
21 attacker(rand_ms_555).

23 2. Using the function CHALLENGE the attacker may obtain CHALLENGE.
24 attacker(CHALLENGE).

26 3. By 2, the attacker may know CHALLENGE.
27 By 1, the attacker may know rand_ms_555.
28 Using the function 2-tuple the attacker may obtain (CHALLENGE,
       rand_ms_555).
29 attacker((CHALLENGE,rand_ms_555)).

31 4. The message (CHALLENGE,rand_ms_555) that the attacker may have by
        3 may be received at input {6}.
```

```
32 So event endMS(imsi,a8(rand_ms_555,ki_560)) may be executed at {9}
       in session endsid_557.
33 end(endsid_557,endMS(imsi,a8(rand_ms_555,ki_560))).

35 A more detailed output of the traces is available with
36   set traceDisplay = long.

38 new imsi_ms creating imsi_ms_562 at {2} in copy a

40 new ki creating ki_563 at {3} in copy a

42 insert keys(imsi_ms_562,ki_563) at {4} in copy a

44 out(pubChannel, (ID,imsi_ms_562)) at {5} in copy a

46 in(pubChannel, (CHALLENGE,a_561)) at {6} in copy a

48 event(endMS(imsi_ms_562,a8(a_561,ki_563))) at {9} in copy a

50 The event endMS(imsi_ms_562,a8(a_561,ki_563)) is executed in session
       a.

52 A trace has been found.

54 RESULT event(endMS(x1,x2)) ==> event(begMS(x1,x2)) is false.

56 RESULT event(endMS(x1_550,x2_551)) ==> event(begMS(x1_550,x2_551))
       is false.)
```

Appendix B

3G and 4G Analysis

B.1 ProVerif Model of Modifiable Multiple IMSIs

```
1  (* Communication Channels *)
2  free pubChannel: channel.
3  free secureChannel: channel [private].

5  (* Type declaration *)
6          type key.
7          type cipherKey.
8          type integrityKey.
9          type anonymityKey.
10         type maskKey.
11         type nonce.
12         type mac.
13         type ident.
14         type resp.
15         type msgHdr.

17         const ID: msgHdr.
18         const AV_REQ: msgHdr.
19         const AV: msgHdr.
20         const CHALLENGE: msgHdr.
21         const RES: msgHdr.

23 (* Cryptographic functions *)
24         fun f1(bitstring, mac, bitstring, nonce, key): mac.
25         fun f11(bitstring, key): mac.
26         fun f2(mac, bitstring, nonce, key): resp.
```

```
27          fun f3(mac, bitstring, nonce, key): cipherKey.
28          fun f4(mac, bitstring, nonce, key): integrityKey.
29          fun f5(mac, bitstring, nonce, key): anonymityKey.
30          fun f51(bitstring, key): maskKey.
31          fun aencrypt(bitstring, anonymityKey): bitstring.
32          fun mencrypt(bitstring, maskKey): bitstring.

34  (* Reduction Rules*)
35          reduc forall m: bitstring, k: anonymityKey; adecrypt(
                aencrypt(m, k), k) = m.
36          reduc forall n: bitstring, l: maskKey; mdecrypt(mencrypt(n,
                l), l) = n.

38  (* Store identity and key pair *)
39          table keys(ident, key).

41  (* Private data which secrecy we are interested in *)
42          free sqn: bitstring [private].
43          free msin: bitstring [private].

45  (* Reachability / Secrecy queries *)
46          query attacker(sqn).
47          query attacker(msin).

49  (* Event declaration: To mark events of interest *)
50          event begSN(ident, cipherKey, integrityKey).
51          event endSN(ident, cipherKey, integrityKey).
52          event begUE(ident, cipherKey, integrityKey).
53          event endUE(ident, cipherKey, integrityKey).

55  (* Authentication queries *)

57  (* SN authenticates UE *)
58          query x1: ident, x2: cipherKey, x3: integrityKey; event(
                endSN(x1, x2, x3)) ==> event(begSN(x1, x2, x3)).
59  (* UE authenticates SN *)
60          query x1: ident, x2: cipherKey, x3: integrityKey; event(
                endUE(x1, x2, x3)) ==> event(begUE(x1, x2, x3)).

62  (* Subscriber sub-process *)
63  let processUE=
64          new imsi_ms: ident;
```

```
65          new ki: key;
66          insert keys(imsi_ms, ki);
67          (* MSG 1 Send *)
68          out(pubChannel, (ID, imsi_ms));
69          (* MSG 4 Receive *)
70          in(pubChannel, (=CHALLENGE, SMAC_ms: mac, masked_msin_ms:
                bitstring, r_ms: nonce, enc_sqn_ms: bitstring, mac_ms:
                mac));
71          let ak_ms: anonymityKey = f5(SMAC_ms, masked_msin_ms, r_ms,
                ki) in
72          let sqn_ms: bitstring = adecrypt(enc_sqn_ms, ak_ms) in
73          if f1(sqn_ms, SMAC_ms, masked_msin_ms, r_ms, ki) = mac_ms
                then
74                  let res_ms: resp = f2(SMAC_ms, masked_msin_ms, r_ms,
                        ki) in
75                  let ck_ms: cipherKey = f3(SMAC_ms, masked_msin_ms,
                        r_ms, ki) in
76                  let ik_ms: integrityKey = f4(SMAC_ms, masked_msin_ms
                        , r_ms, ki) in
77                  event endMS(imsi_ms, ck_ms, ik_ms);
78                  event begSN(imsi_ms, ck_ms, ik_ms);
79          (* MSG 5 Send *)
80          out(pubChannel, (RES, res_ms))
81          if f11(sqn_ms, ki) = SMAC_ms then
82                          let ek_ms: maskKey = f51(sqn_ms, ki) in
83                          let msin_ms: bitstring = mdecrypt(
                                masked_msin_ms, ek_ms) in 0.

85  (* Serving Network sub-process *)
86  let processSN=
87          (* MSG 1 Receive *)
88          in(pubChannel, (=ID, imsi_sn: ident));
89          (* MSG 2 Send *)
90          out(secureChannel, (AV_REQ, imsi_sn));
91          (* MSG 3 Receive *)
92          in(secureChannel, (=AV, imsi_hn_sn: ident, SMAC_sn:
                bitstring, masked_msin_sn: bitstring, r_sn: nonce,
                enc_sqn_sn: bitstring, mac_sn: mac, xres_sn: resp, ck_sn:
                 cipherKey, ik_sn: integrityKey));
93          event begMS(imsi_hn_sn, ck_sn, ik_sn);
94          (* MSG 4 Send *)
```

```
95          out(pubChannel, (CHALLENGE, SMAC_sn, masked_msin_sn, r_sn,
                enc_sqn_sn, mac_sn));
96          (* MSG 5 Receive *)
97          in(pubChannel, (=RES, res_sn: resp));
98          if res_sn = xres_sn then
99          event endSN(imsi_hn_sn, ck_sn, ik_sn).

101 (* Home Network sub-process *)
102 let processHN=
103         (* MSG 2 Receive *)
104         in(secureChannel, (=AV_REQ, imsi_hn: ident));
105         get keys(=imsi_hn, ki_hn) in
106         new r_hn: nonce;
107         let SMAC_hn: mac = f11(sqn, ki_hn) in
108         let ek_hn: maskKey = f51(sqn, ki_hn) in
109         let masked_msin_hn: bitstring = mencrypt(msin, ek_hn) in
110         let mac_hn: mac = f1(sqn, SMAC_hn, masked_msin_hn, r_hn,
                ki_hn) in
111         let xres_hn: resp = f2(SMAC_hn, masked_msin_hn, r_hn, ki_hn)
                in
112         let ck_hn: cipherKey = f3(SMAC_hn, masked_msin_hn, r_hn,
                ki_hn) in
113         let ik_hn: integrityKey = f4(SMAC_hn, masked_msin_hn, r_hn,
                ki_hn) in
114         let ak_hn: anonymityKey = f5(SMAC_hn, masked_msin_hn, r_hn,
                ki_hn) in
115         let enc_sqn_hn: bitstring = aencrypt(sqn, ak_hn) in
116         (* MSG 3 Send *)
117         out(secureChannel, (AV, imsi_hn, SMAC_hn, masked_msin_hn,
                r_hn, enc_sqn_hn, mac_hn, xres_hn, ck_hn, ik_hn)).

119 (* Main process *)
120                 process
121                         ((!processMS) | processSN | processHN)
```

B.2 ProVerif Model of Robust Pseudo-IMSIs

```
1  (* Communication Channels *)
2  free pubChannel: channel.
3  free secureChannel: channel [private].

5  (* Type declaration *)
6  type key.
7  type cipherKey.
8  type integrityKey.
9  type anonymityKey.
10 type maskKey.
11 type nonce.
12 type mac.
13 type ident.
14 type resp.
15 type msgHdr.

17 const ID: msgHdr.
18 const AV_REQ: msgHdr.
19 const AV: msgHdr.
20 const CHALLENGE: msgHdr.
21 const RES: msgHdr.
22 const ERROR: msgHdr.

24 (* Cryptographic functions *)
25 fun f1(bitstring, bitstring, bitstring, nonce, key): mac.
26   (* Function to compute MAC used in the error token *)
27 fun f12(bitstring, ident, key): mac.
28 fun f2(bitstring, bitstring, nonce, key): resp.
29 fun f3(bitstring, bitstring, nonce, key): cipherKey.
30 fun f4(bitstring, bitstring, nonce, key): integrityKey.
31 fun f5(bitstring, bitstring, nonce, key): anonymityKey.
32 fun f52(bitstring, key): maskKey.
33 fun f53(bitstring, key): maskKey.
34 fun aencrypt(bitstring, anonymityKey): bitstring.
35 fun mencrypt(bitstring, maskKey): bitstring.

37 (* Reduction Rules*)
38 reduc forall m: bitstring, k: anonymityKey; adecrypt(aencrypt(m, k),
       k) = m.
```

```
39 reduc forall n: bitstring, l: maskKey; mdecrypt(mencrypt(n, l), l) =
       n.

41 (* Store relevant data *)
42 table keys(ident, key).
43 table rids(bitstring, ident).

45 (* Private data which secrecy we are interested in *)
46 free sqn: bitstring [private].
47 free tid: bitstring [private].
48 free success: bool [private].

50 (* Reachability / Secrecy queries *)
51 query attacker(sqn).
52 query attacker(tid).

54 (* Event declaration: To mark events of interest *)
55 event begRecoveryHN(ident, bitstring).
56 event endRecoveryHN(ident, bitstring).
57 event begSN(ident, cipherKey, integrityKey).
58 event endSN(ident, cipherKey, integrityKey).
59 event begUE(ident, cipherKey, integrityKey).
60 event endUE(ident, cipherKey, integrityKey).

62 (* Authentication queries *)
63         (* SN authenticates MS *)
64 query x1: ident, x2: cipherKey, x3: integrityKey; event(endSN(x1, x2
       , x3)) ==> event(begSN(x1, x2, x3)).
65         (* MS authenticates SN *)
66 query x1: ident, x2: cipherKey, x3: integrityKey; event(endUE(x1, x2
       , x3)) ==> event(begUE(x1, x2, x3)).

68 (* Correctness of pseudo-IMSI synchronisation recovery query*)
69         (* HN Confirms that the rid is sent by a valid subscriber *)
70 query x1: ident, x2: bitstring; event(endRecoveryHN(x1, x2)) ==>
       event(begRecoveryHN(x1, x2)).

72 (* Subscriber sub-process *)
73 let processUE=
74  new imsi_ms: ident;
75  new ki: key;
76  insert keys(imsi_ms, ki);
```

```
77 (* MSG 1 Send *)
78  out(pubChannel, (ID, imsi_ms));
79 (* MSG 4 Receive *)
80  in(pubChannel, (=CHALLENGE, masked_rid_ms: bitstring, masked_tid_ms
        : bitstring, r_ms: nonce, enc_sqn_ms: bitstring, mac_ms: mac));
81  new d_mac: mac;
82  new d_rid: bitstring;
83  let ak_ms: anonymityKey = f5(masked_rid_ms, masked_tid_ms, r_ms, ki
        ) in
84  let sqn_ms: bitstring = adecrypt(enc_sqn_ms, ak_ms) in
85  if f1(sqn_ms, masked_rid_ms, masked_tid_ms, r_ms, ki) = mac_ms then
         (
86   let res_ms: resp = f2(masked_rid_ms, masked_tid_ms, r_ms, ki) in
87   let ck_ms: cipherKey = f3(masked_rid_ms, masked_tid_ms, r_ms, ki)
        in
88   let ik_ms: integrityKey = f4(masked_rid_ms, masked_tid_ms, r_ms,
        ki) in
89   let success = true in
90         event endMS(imsi_ms, ck_ms, ik_ms);
91         event begSN(imsi_ms, ck_ms, ik_ms);
92 (* MSG 5a Send *)
93   out(pubChannel, (RES, success, res_ms, d_rid, d_mac)))
94  else (
95 (* Generate MAC-error and sent MSG 5b *)
96   new d_res: resp;
97   let success = false in
98   let ek2_ms: maskKey = f53(sqn_ms, ki) in
99   let rid_ms: bitstring = mdecrypt(masked_rid_ms, ek2_ms) in
100  let mac_m_ms: mac = f12(rid_ms, imsi_ms, ki) in
101 (* MSG 5b Send *)
102        event begRecoveryHN(imsi_ms, rid_ms);
103  out(pubChannel, (RES, success, d_res, rid_ms, mac_m_ms))).

105 (* Serving Network sub-process *)
106 let processSN=
107 (* MSG 1 Receive *)
108  in(pubChannel, (=ID, imsi_sn: ident));
109 (* MSG 2 Send *)
110  out(secureChannel, (AV_REQ, imsi_sn));
111 (* MSG 3 Receive *)
112  in(secureChannel, (=AV, imsi_hn_sn: ident, masked_rid_sn: bitstring
        , masked_tid_sn: bitstring, r_sn: nonce, enc_sqn_sn: bitstring,
```

```
        mac_sn: mac, xres_sn: resp, ck_sn: cipherKey, ik_sn:
        integrityKey));
113         event begMS(imsi_hn_sn, ck_sn, ik_sn);
114 (* MSG 4 Send *)
115  out(pubChannel, (CHALLENGE, masked_rid_sn, masked_tid_sn, r_sn,
        enc_sqn_sn, mac_sn));
116 (* MSG 5a/5b Receive *)
117  in(pubChannel, (=RES, success_sn: bool, res_sn: resp, rid_sn:
        bitstring, mac_m_sn: mac));
118  if success_sn = true then
119   ( if res_sn = xres_sn then event endSN(imsi_hn_sn, ck_sn, ik_sn))
120   else
121 (* MSG 6 Send*)
122    out(secureChannel, (ERROR, imsi_hn_sn, rid_sn, mac_m_sn)).

124 (* Home Network sub-process *)
125 let processHN=
126 (* MSG 2 Receive *)
127  in(secureChannel, (=AV_REQ, imsi_hn: ident));
128  get keys(=imsi_hn, ki_hn) in
129  new r_hn: nonce;
130  new rid_hn: bitstring;
131  insert rids(rid_hn, imsi_hn);
132  let ek2_hn: maskKey = f53(sqn, ki_hn) in
133  let masked_rid_hn: bitstring = mencrypt(rid_hn, ek2_hn) in
134  let ek_hn: maskKey = f52(sqn, ki_hn) in
135  let masked_tid_hn: bitstring = mencrypt(tid, ek_hn) in
136  let mac_hn: mac = f1(sqn, masked_rid_hn, masked_tid_hn, r_hn, ki_hn
        ) in
137  let xres_hn: resp = f2(masked_rid_hn, masked_tid_hn, r_hn, ki_hn)
        in
138  let ck_hn: cipherKey = f3(masked_rid_hn, masked_tid_hn, r_hn, ki_hn
        ) in
139  let ik_hn: integrityKey = f4(masked_rid_hn, masked_tid_hn, r_hn,
        ki_hn) in
140  let ak_hn: anonymityKey = f5(masked_rid_hn, masked_tid_hn, r_hn,
        ki_hn) in
141  let enc_sqn_hn: bitstring = aencrypt(sqn, ak_hn) in
142 (* MSG 3 Send *)
143   out(secureChannel, (AV, imsi_hn, masked_rid_hn, masked_tid_hn,
         r_hn, enc_sqn_hn, mac_hn, xres_hn, ck_hn, ik_hn));
144 (* MSG 6 Receive *)
```

```
145  in(secureChannel, (=ERROR, imsi_hn_s: ident, r_rid_hn: bitstring,
       mac_m_hn: mac));
146  get rids(=r_rid_hn, imsi_hn2) in
147  if imsi_hn_s = imsi_hn2 then
148    get keys(=imsi_hn2, ki_hn2) in
149    if f12(r_rid_hn, imsi_hn2, ki_hn2) = mac_m_hn then
150  (* AUTM token verification successful *)
151                  event endRecoveryHN(imsi_hn2, r_rid_hn).

153  (* Main process *)
154         process
155                  ((!processMS) | processSN | processHN)
```

B.3 ProVerif Output of Model Execution

B.3.1 Modifiable Multiple IMSIs

```
(* Verification of Authentication Property: SN Authentication *)

-- Query event(endMS(x1,x2,x3)) ==> event(begMS(x1,x2,x3))
Completing...
Starting query event(endMS(x1,x2,x3)) ==> event(begMS(x1,x2,x3))
goal reachable: begin(begMS(imsi_ms[!1 = @sid_1933],f3(f11(sqn[],ki
    [!1 = @sid_1933]),mencrypt(msin[],f51(sqn[],ki[!1 = @sid_1933])),
    r_hn[ki_hn = ki[!1 = @sid_1933],imsi_hn = imsi_ms[!1 = @sid_1933
    ]],ki[!1 = @sid_1933]),f4(f11(sqn[],ki[!1 = @sid_1933]),mencrypt(
    msin[],f51(sqn[],ki[!1 = @sid_1933])),r_hn[ki_hn = ki[!1 =
    @sid_1933],imsi_hn = imsi_ms[!1 = @sid_1933]],ki[!1 = @sid_1933])
    )) -> end(endMS(imsi_ms[!1 = @sid_1933],f3(f11(sqn[],ki[!1 =
    @sid_1933]),mencrypt(msin[],f51(sqn[],ki[!1 = @sid_1933])),r_hn[
    ki_hn = ki[!1 = @sid_1933],imsi_hn = imsi_ms[!1 = @sid_1933]],ki
    [!1 = @sid_1933]),f4(f11(sqn[],ki[!1 = @sid_1933]),mencrypt(msin
    [],f51(sqn[],ki[!1 = @sid_1933])),r_hn[ki_hn = ki[!1 = @sid_1933
    ],imsi_hn = imsi_ms[!1 = @sid_1933]],ki[!1 = @sid_1933])))

RESULT event(endMS(x1,x2,x3)) ==> event(begMS(x1,x2,x3)) is true.

(* Verification of Authentication Property: MS Authentication *)

-- Query event(endSN(x1_1937,x2_1938,x3_1939)) ==> event(begSN(
    x1_1937,x2_1938,x3_1939))
Completing...
Starting query event(endSN(x1_1937,x2_1938,x3_1939)) ==> event(begSN
    (x1_1937,x2_1938,x3_1939))
goal reachable: begin(begSN(imsi_ms[!1 = @sid_3452],f3(f11(sqn[],ki
    [!1 = @sid_3452]),mencrypt(msin[],f51(sqn[],ki[!1 = @sid_3452])),
    r_hn[ki_hn = ki[!1 = @sid_3452],imsi_hn = imsi_ms[!1 = @sid_3452
    ]],ki[!1 = @sid_3452]),f4(f11(sqn[],ki[!1 = @sid_3452]),mencrypt(
    msin[],f51(sqn[],ki[!1 = @sid_3452])),r_hn[ki_hn = ki[!1 =
    @sid_3452],imsi_hn = imsi_ms[!1 = @sid_3452]],ki[!1 = @sid_3452])
    )) -> end(endSN(imsi_ms[!1 = @sid_3452],f3(f11(sqn[],ki[!1 =
    @sid_3452]),mencrypt(msin[],f51(sqn[],ki[!1 = @sid_3452])),r_hn[
    ki_hn = ki[!1 = @sid_3452],imsi_hn = imsi_ms[!1 = @sid_3452]],ki
    [!1 = @sid_3452]),f4(f11(sqn[],ki[!1 = @sid_3452]),mencrypt(msin
    [],f51(sqn[],ki[!1 = @sid_3452])),r_hn[ki_hn = ki[!1 = @sid_3452
```

```
    ],imsi_hn = imsi_ms[!1 = @sid_3452]],ki[!1 = @sid_3452])))

17 RESULT event(endSN(x1_1937,x2_1938,x3_1939)) ==> event(begSN(x1_1937
       ,x2_1938,x3_1939)) is true.

19 (* Verification of Secrecy Property *)

21 -- Query not attacker(msin[])
22 Completing...
23 Starting query not attacker(msin[])

25 RESULT not attacker(msin[]) is true.

27 -- Query not attacker(sqn[])
28 Completing...
29 Starting query not attacker(sqn[])

31 RESULT not attacker(sqn[]) is true.
```

B.3.2 Robust Pseudo-IMSIs

```
1  (* Verification of Authentication Property: SN Authentication *)

3  -- Query event(endMS(x1,x2,x3)) ==> event(begMS(x1,x2,x3))
4  Completing...
5  Starting query event(endMS(x1,x2,x3)) ==> event(begMS(x1,x2,x3))
6  goal reachable: begin(begMS(imsi_ms[!1 = @sid_2887],f3(mencrypt(
       rid_hn[ki_hn = ki[!1 = @sid_2887],imsi_hn = imsi_ms[!1 =
       @sid_2887]],f53(sqn[],ki[!1 = @sid_2887])),mencrypt(tid[],f52(sqn
       [],ki[!1 = @sid_2887])),r_hn[ki_hn = ki[!1 = @sid_2887],imsi_hn =
        imsi_ms[!1 = @sid_2887]],ki[!1 = @sid_2887]),f4(mencrypt(rid_hn[
       ki_hn = ki[!1 = @sid_2887],imsi_hn = imsi_ms[!1 = @sid_2887]],f53
       (sqn[],ki[!1 = @sid_2887])),mencrypt(tid[],f52(sqn[],ki[!1 =
       @sid_2887])),r_hn[ki_hn = ki[!1 = @sid_2887],imsi_hn = imsi_ms[!1
        = @sid_2887]],ki[!1 = @sid_2887]))) -> end(endMS(imsi_ms[!1 =
       @sid_2887],f3(mencrypt(rid_hn[ki_hn = ki[!1 = @sid_2887],imsi_hn
       = imsi_ms[!1 = @sid_2887]],f53(sqn[],ki[!1 = @sid_2887])),
       mencrypt(tid[],f52(sqn[],ki[!1 = @sid_2887])),r_hn[ki_hn = ki[!1
       = @sid_2887],imsi_hn = imsi_ms[!1 = @sid_2887]],ki[!1 = @sid_2887
       ]),f4(mencrypt(rid_hn[ki_hn = ki[!1 = @sid_2887],imsi_hn =
       imsi_ms[!1 = @sid_2887]],f53(sqn[],ki[!1 = @sid_2887])),mencrypt(
       tid[],f52(sqn[],ki[!1 = @sid_2887])),r_hn[ki_hn = ki[!1 =
       @sid_2887],imsi_hn = imsi_ms[!1 = @sid_2887]],ki[!1 = @sid_2887])
       ))

8  RESULT event(endMS(x1,x2,x3)) ==> event(begMS(x1,x2,x3)) is true.

10 (* Verification of Authentication Property: MS Authentication *)

12 -- Query event(endSN(x1_2891,x2_2892,x3_2893)) ==> event(begSN(
       x1_2891,x2_2892,x3_2893))
13 Completing...
14 Starting query event(endSN(x1_2891,x2_2892,x3_2893)) ==> event(begSN
       (x1_2891,x2_2892,x3_2893))
15 goal reachable: begin(begSN(imsi_ms[!1 = @sid_5231],f3(mencrypt(
       rid_hn[ki_hn = ki[!1 = @sid_5231],imsi_hn = imsi_ms[!1 =
       @sid_5231]],f53(sqn[],ki[!1 = @sid_5231])),mencrypt(tid[],f52(sqn
       [],ki[!1 = @sid_5231])),r_hn[ki_hn = ki[!1 = @sid_5231],imsi_hn =
        imsi_ms[!1 = @sid_5231]],ki[!1 = @sid_5231]),f4(mencrypt(rid_hn[
       ki_hn = ki[!1 = @sid_5231],imsi_hn = imsi_ms[!1 = @sid_5231]],f53
       (sqn[],ki[!1 = @sid_5231])),mencrypt(tid[],f52(sqn[],ki[!1 =
```

```
@sid_5231])),r_hn[ki_hn = ki[!1 = @sid_5231],imsi_hn = imsi_ms[!1
 = @sid_5231]],ki[!1 = @sid_5231]))) -> end(endSN(imsi_ms[!1 =
@sid_5231],f3(mencrypt(rid_hn[ki_hn = ki[!1 = @sid_5231],imsi_hn
= imsi_ms[!1 = @sid_5231]],f53(sqn[],ki[!1 = @sid_5231])),
mencrypt(tid[],f52(sqn[],ki[!1 = @sid_5231])),r_hn[ki_hn = ki[!1
= @sid_5231],imsi_hn = imsi_ms[!1 = @sid_5231]],ki[!1 = @sid_5231
]),f4(mencrypt(rid_hn[ki_hn = ki[!1 = @sid_5231],imsi_hn =
imsi_ms[!1 = @sid_5231]],f53(sqn[],ki[!1 = @sid_5231])),mencrypt(
tid[],f52(sqn[],ki[!1 = @sid_5231])),r_hn[ki_hn = ki[!1 =
@sid_5231],imsi_hn = imsi_ms[!1 = @sid_5231]],ki[!1 = @sid_5231])
))

RESULT event(endSN(x1_2891,x2_2892,x3_2893)) ==> event(begSN(x1_2891
,x2_2892,x3_2893)) is true.

(* Verification of Correctness of the Pseudo-IMSI synchronisation
Recovery *)

-- Query event(endRecoveryHN(x1_5235,x2_5236)) ==> event(
begRecoveryHN(x1_5235,x2_5236))
Completing...
Starting query event(endRecoveryHN(x1_5235,x2_5236)) ==> event(
begRecoveryHN(x1_5235,x2_5236))
goal reachable: begin(begRecoveryHN(imsi_ms[!1 = @sid_7636],rid_hn[
ki_hn = ki[!1 = @sid_7636],imsi_hn = imsi_ms[!1 = @sid_7636]]))
-> end(endRecoveryHN(imsi_ms[!1 = @sid_7636],rid_hn[ki_hn = ki[!1
 = @sid_7636],imsi_hn = imsi_ms[!1 = @sid_7636]]))

RESULT event(endRecoveryHN(x1_5235,x2_5236)) ==> event(begRecoveryHN
(x1_5235,x2_5236)) is true.

(* Verification of Secrecy Property *)

-- Query not attacker(tid[])
Completing...
Starting query not attacker(tid[])

RESULT not attacker(tid[]) is true.

-- Query not attacker(sqn[])
Completing...
Starting query not attacker(sqn[])
```

```
40 RESULT not attacker(sqn[]) is true.
```